Visual Analysis of Neuromuscular Junctions

Vom Fachbereich Informatik der
Technischen Universität Kaiserslautern
zur Verleihung des akademischen Grades
Doktor der Naturwissenschaften (Dr. rer. nat.)

genehmigte Dissertation

von

Julia Portl

Datum der wissenschaftlichen Aussprache: 08.12.2016

Dekan des Fachbereichs: Prof. Dr. Klaus Schneider
Erster Berichterstatter: Prof. Dr. Heike Leitte
Zweiter Berichterstatter: Prof. Dr. Hans Hagen

D 386

Abstract

Synapses play a central role in the information propagation in the nervous system. A better understanding of synaptic structures and processes is vital for advancing nervous disease research. This work is part of an interdisciplinary project that aims at the quantitative examination of components of the *neuromuscular junction*, a synaptic connection between a neuron and a muscle cell.

The research project is based on image stacks picturing neuromuscular junctions captured by modern electron microscopes, which permit the rapid acquisition of huge amounts of image data at a high level of detail. The large amount and sheer size of such microscopic data makes a direct visual examination infeasible, though.

This thesis presents novel problem-oriented interactive visualization techniques that support the segmentation and examination of neuromuscular junctions.

First, I introduce a structured data model for segmented surfaces of neuromuscular junctions to enable the computational analysis of their properties. However, surface segmentation of neuromuscular junctions is a very challenging task due to the extremely intricate character of the objects of interest. Hence, such problematic segmentations are often performed manually by non-experts and thus requires further inspection.

With *NeuroMap*, I develop a novel framework to support proofreading and correction of three-dimensional surface segmentations. To provide a clear overview and to ease navigation within the data, I propose the *surface map*, an abstracted two-dimensional representation using key features of the surface as landmarks. These visualizations are augmented with information about automated segmentation error estimates. The framework provides intuitive and interactive data correction mechanisms, which in turn permit the expeditious creation of high-quality segmentations.

While analyzing such segmented synapse data, the formulation of specific research questions is often impossible due to missing insight into the data. I address this problem by designing a generic parameter space for segmented structures from biological image data. Furthermore, I introduce a graphical interface to aid its exploration, combining both parameter selection as well as data representation.

Zusammenfassung

Ein tiefgreifendes Verständnis von Aufbau und Abläufen an Synapsen ist unumgänglich für Fortschritte in der Erforschung von Nervenerkrankungen. Die vorliegende Arbeit ist Teil eines interdisziplinären Projektes, dessen Ziel es ist, Synapsen zwischen Nerven- und Muskelzellen—*neuromuscular junctions*, kurz *NMJ*—quantitativ zu untersuchen.

Die Datengrundlage hierbei bilden Bildstapel von NMJ, die mittels moderner Elektronenmikroskopie aufgenommen wurden, welche die rasche Aufnahme von riesigen Datenmengen mit hohem Detailreichtum erlauben. Durch ihre schiere Menge und Größe ist eine direkte optische Analyse der Bilddaten jedoch nicht praktikabel.

In dieser Arbeit stelle ich neuartige problemorientierte und interaktive Visualisierungstechniken vor, die der Segmentierung und Untersuchung von NMJ dienen.

Dazu führe ich zuerst ein strukturiertes Datenmodell für segmentierte Oberflächen von NMJ ein, um die Analyse der Eigenschaften von Bestandteilen der NMJ zu ermöglichen. Die Segmentierung von NMJ gestaltet sich wegen der Filigranität ihrer Bestandteile jedoch als sehr schwierig. Deshalb werden solche Segmentierungen oftmals manuell erzeugt, aus Kostengründen allerdings nicht von Fachleuten, weshalb eine anschließende Überprüfung durch Spezialisten notwendig ist.

Daher habe ich die Software *NeuroMap* entwickelt, die das Korrekturlesen und die Korrektur von dreidimensionalen Oberflächensegmentierungen unterstützt. Diese enthält eine zweidimensionale Darstellung von charakteristische Eigenschaften der segmentierten Oberfläche, welche zugleich als Übersicht über die segmentierten Bilddaten, sowie auch als Navigationshilfe innerhalb der Daten dient. Die Visualisierungen enthalten automatisch geschätzte Informationen über die Qualität der Segmentierungen. Des Weiteren bietet die Software das sowohl intuitive als auch interaktive Korrigieren von Oberflächensegmentierungen an, was eine zügige Erstellung von qualitativ hochwertigen Segmentierungen ermöglicht.

Bei der Analyse solcher segmentierter Daten ist die Formulierung von spezifischen Fragestellungen nicht möglich, da der Einblick in die Daten fehlt. Ich stelle für dieses Problem die Entwicklung eines generischen Parameterraums für segmentierte Objekte aus biologischen Bilddaten vor. Des Weiteren präsentiere ich eine graphische Oberfläche, die Parameterauswahl und Datenrepräsentation vereinigt und bei der explorativen Analyse des zu untersuchenden Parameterraumes behilflich ist.

Danksagung

An dieser Stelle möchte ich all den lieben Menschen danken, die mich auf dem Weg durch meine Promotion begleitet haben und die es mir ermöglicht haben, diesen Weg zu gehen.

Mein besonderer Dank gilt hierbei Heike Leitte für das Vertrauen, das sie mir in all den Jahren entgegengebracht hat. Sie ist mir ein Vorbild und hat mich, seit ich sie kenne, immer unterstützt, ermutigt und an mich geglaubt. Ich danke ihr herzlich, dass sie sich immer für mich eingesetzt hat und dass ich sehr viel von ihr lernen konnte, sowohl fachlich und als auch persönlich.

Auch möchte ich Hans Hagen mit seiner fürsorglichen und mitreißenden Art aus vollem Herzen danken. Sein unermüdlicher Einsatz für mein Wohl hat mich tief beeindruckt und sehr gerührt.

Ich möchte mich ganz besonders bei Ira Mang für unsere angenehme Zusammenarbeit, ihre Offenheit und Geduld bedanken. Ich konnte mir ihrer Unterstützung stets sicher sein und schätze ihre gelassene und ruhige Art sehr. Mein Dank gilt ebenfalls Rasmus Schröder für die großzügige fachliche und finanzielle Unterstützung. Auch danken möchte ich Markus Reischl und Johannes Stegmaier für ihre hilfsbereite und unkomplizierte Zusammenarbeit.

Ich danke Bastian Rieck mit dem ich mir sechs Jahre ein Büro teilte und dessen tägliche Gesellschaft ich sehr vermissen werde. Er stand mir mit seinen unglaublichen Wissensschatz und seiner bedingungslosen Hilfsbereitschaft immer mit Rat und Tat zur Seite. Zu meinen engsten Kollegen gehören auch Andreas Beyer und Bartosz Bogacz, die mir im Laufe der Jahre zu sehr wertvollen Freunden geworden sind und die immer bereit für fachliche und persönliche Gespräche waren. Ich danke ihnen für die tolle gemeinsame Zeit.

Mein Dank gilt auch Susanne Krömker, Michael Winckler, Filip Sadlo, Christian Heine und Hubert Mara, die mich während und auch schon vor meiner Promotion bei meinem akademischen Werdegang sehr unterstützt haben. Außerdem danke ich der *HGS Mathcomp* für die finanzielle Unterstützung sowie für ihre vielzähligen Möglichkeiten zur Weiterbildung in fachlichen, geselligen und sportlichen Bereichen.

Ich möchte mich bei den Mitarbeiter*innen der Sekretariate und Verwaltung des IWR und der HGS, sowie der TU Kaiserslautern, insbesondere des FB Informatik,

bedanken. Sie leisten allesamt tolle Arbeit und machen mit viel Geduld und Engagement immer wieder Unmögliches möglich.

Ich danke allen aktuellen und ehemaligen Kolleg*innen und Student*innen des 5. Stockwerks des IWR als auch des Mathematikons. Ich bin dankbar, dass ich einen Großteil meiner Studienzeit und Promotion mit solch aufgeschlossenen Menschen in einer gelösten Atmosphäre verbringen durfte. Besonders erwähnen und danken möchte ich hierbei Markus Ridinger, Elfriede Friedmann und Martin Neissen. Großer Dank geht auch an meine Kolleg*innen in Kaiserslautern für das herzliche Willkommen und ihre freundliche und vorbehaltlose Aufnahme meiner Person in ihre Mitte.

Besonderer Dank gilt meiner Familie, die mich jederzeit und auf jede nur erdenkliche Weise unterstützen. Außerdem danke ich Angelika Fröhlich und all den lieben Menschen, die ich meine Freunde nennen darf.

CONTENTS

Dedicated to my parents

1 Introduction

This chapter gives an overview of the research project in which this work is embedded. It introduces the individuals of the project team as well as their respective areas of competence. Moreover, it portrays the relevance of the application field as well as the contributions of this thesis.

1.1 Context of this Work

The work that is presented in this thesis is part of an interdisciplinary collaborative project with team members from three different disciplines: biology, image processing, and scientific visualization. The objective of this project is a better understanding of the neuromuscular junction (NMJ), which is the synaptic connection between a nerve cell and a muscle cell. The research is based on microscopic image data of muscle tissue and nerve cells. To this end, volumes of muscle tissue are captured using electron microscopes, resulting in stacks of vast amounts of high-resolution image data. In these images stacks, the components of an NMJ are identified. Each one of them is subsequently described mathematically within a segmentation process. These components are used for a further in-depth analysis—both of a statistical and a visual kind—of the relevant objects, from which it is possible to draw valid biological conclusions.

1.1.1 Collaborations

Our partners from biology are Ira V. Mang and Rasmus R. Schröder from the *Cryo-Electron Microscopy Group* of *BioQuant* at Heidelberg University. They are experts in microscopic image acquisition and for the anatomy and physiology of tissue and cells.

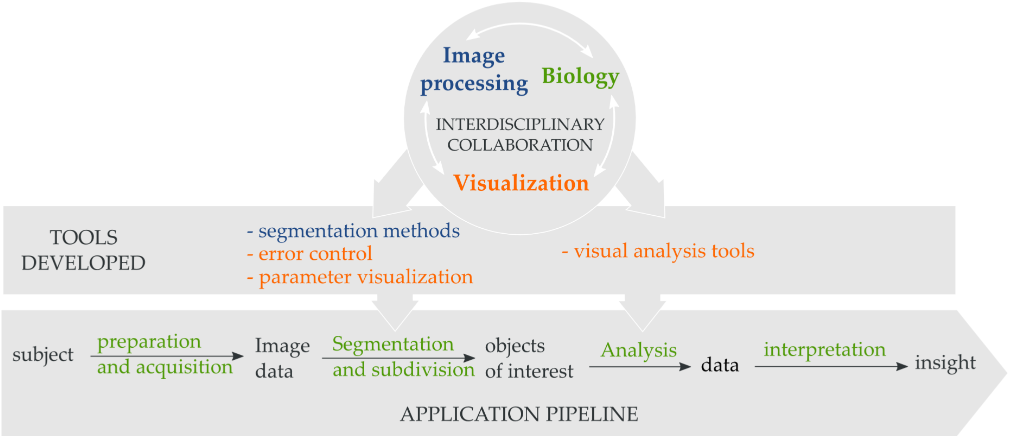


FIGURE 1.1: The interaction of the project parts and partners. Colors indicate affiliations and responsibilities. Top: collaboration of the three research teams. Bottom: research pipeline of the biological application domain. Middle: tools and methods that are developed by the respective competence teams in close collaboration with the partners to serve the examination of the biological research questions.

Markus Reischl, Johannes Stegmaier, Ralf Mikut, and Nico Peter from the *Karlsruhe Institute of Technology (KIT)* are specialists in image processing. Their algorithms identify the objects of interest in raw microscopic image data.

The third team consists of Heike Leitte, the head of the *Visual Information Analysis Group* at Technical University Kaiserslautern, and Julia Portl, the author of this thesis. We both were affiliated with the *Interdisciplinary Center of Scientific Computing (IWR)* at Heidelberg University for the first period of the project. In this thesis, I present approaches that we have been developing over the last years, from the field of scientific visualization to tackle this project's challenging complex of biological problems.

The work was financed by the *HGS MathComp*, TU Kaiserslautern (via the *Professorinnen Programm* of BMBF[1]), the *HeiKa Research Partnership* (project title "Feedback-based Segmentation and Visualisation of complex 3D-data-sets"), and by the BMBF-funded projects *NanoCombine* (#13N11401) and *MorphiQuant-3D* (#13GW0044).

---

[1] Bundesministerium für Bildung und Forschung (Federal Ministry of Education and Research of the Federal Republic of Germany)

1.1.2 Role of Scientific Visualization

The visualization component becomes important at two crucial parts of the project, namely the segmentation process and the analysis part (Figure 1.1). This allows us to approach the topic from two different perspectives. First of all, we offer a complete framework for achieving high-quality segmentations. For this context, we developed an abstract representation for complex 3D structures that conserves local proximity, similar to the way a subway map preserves the relative location of places. Moreover, we provide support for the structured and comparative analysis of numerical properties of the component models. The formulation of specific research questions pertaining to the data is often impossible due to missing knowledge. This necessitates the exploration of different aspects of the data in order to obtain an overview. Furthermore, there is a dire need for tools that permit analysts to rapidly explore properties of their data while at the same time preventing superfluous queries.

1.2 Motivation

Synapses play a central role in the information propagation within the nervous system. Errors in their structure or the biochemical processes therein may cause severe problems such as nervous diseases. If synaptic structures and processes were better understood, nervous disease research would greatly benefit. Examples are the fundamental research on Alzheimer's and Parkinson's disease as well as on Duchenne muscular dystrophy, a severe inheritable genetic disorder that leads to an early death.

As synapses in the brain are difficult to access and image, scientists investigate NMJs instead, which are very similar to synapses in the brain, but easier to access and study [1]. A thorough understanding of the biological background of NMJs is crucial for the successful development of techniques that support the investigation of real-world questions and research problems in biology.

Although the NMJ has already been widely researched and a good general understanding of the processes at the NMJ exists, numerous questions concerning the structure of the NMJ and its processes have so far not been answered satisfactorily.

1.2.1 Open Biological Research Questions

For the transmission of a nerve impulse, the axon terminal of the neuron releases a chemical neurotransmitter into the synaptic cleft between the neuron and the muscle cell. Receptors in the membrane of the muscle cell, in turn, bind these transmitters, thereby causing the muscle fiber to contract.

The *junctional folds* are commonly considered to constitute the most signficant features of the NMJ: at regions where the receptors for neurotransmitters are located, the muscle membrane forms characteristic folds that embed the axon terminal of the neuron. The highly-complex geometrical structure of the muscle cell membrane is to a large extent due to these folds.

In current literature, very few models of the junctional folds exist. Stiles et al. [2, 3] are among the few groups to have extracted the membrane from electron microscopy data in order to simulate the diffusion of neurotransmitters in NMJs [4]. However, these approaches are limited to only one single reconstruction and merely comprise small portions of the entire structure. This is due to the fact that—even if supported algorithmically—the reconstruction of the membrane is a laborious and time-consuming task. A representative collection of reliable geometrical models of the NMJs, however, are a crucial component for handling the following research topics.

A central question is whether or to what extent the fold structure influences biochemical processes at the synapses. Diseases such as muscular dystrophies and myasthenic syndromes are usually accompanied by alteration of the NMJ morphology, which is—as of now—only described in a qualitative manner in literature. However, the coherency between the altered membrane's ultrastructure and the muscle function and disease symptoms needs to be explored quantitatively. For this purpose, the differences in shape and topology of the junctional folds between a healthy and dystrophic musculature and nervous system have to be evaluated. In order to derive viable results, not only different groups have to be compared, but also the inter-group variability has to be studied.

Another research topic concerns the migration of receptors. The nicotinic acetylcholine receptors (AChRs) that are located in the muscle membrane, have a half-life of about eight days. New receptors emerge in peripheric regions of the NMJ and are later moved to more central locations. The details of this transport process are still largely unknown. It is generally believed that they are delivered using vesicular transport processes. Mang (née Röder) et al. [5] describe *myosin Va* and *protein kinase A* to function cooperating in recycling AChRs. Even though large time-dependent data series of migrating AChRs cannot be obtained due to imaging limitations (either the resolution is not good enough, or the probe has to be dissected and fixated), AChRs can be labeled at different points in time [6].

This leads to the question whether the morphology of the NMJ behaves differently according to regions. This concerns particularly variations in shape of the junctional folds. Tackling this problem requires examining of volumes that comprise entire NMJs. However, such collections of reconstructions of the muscle cell's membrane are not available yet.

1.2.2 The Benefits of Visualization

Modern electron microscopes permit for the rapid acquisition of huge amounts of image data each of which features a high level of detail. Especially in the life sciences, these images provide an unprecedented level of new insight, enabling scientists to study the fundamental structure and organization of organisms on a subcellular level. Moreover, multi-resolution imaging facilitates the acquisition of data on multiple levels of detail which in turn simplifies the imaging of smallest structures at a resolution of down to 0.1 nm while perceiving on coarser resolutions the larger context.

Despite—or because of—scientists having now access to enormous amounts of data, the analysis of this data turned out to be more difficult than ever before. The large amounts and sheer size of such microscopic data makes a direct visual examination infeasible. While statistics provide summary views of the data, they ultimately lack in specific information about, e.g. abnormalities in feature distributions in relation to the spatial location they occur. Moreover, computational measurements of biological features can lead to enormous amounts of numerical data, which is difficult to grasp. Here, visualization is capable of communicating both numerical data combined with spatial information.

1.3 Contributions of this Thesis

Within this interdisciplinary project I developed problem-oriented interactive visualization techniques to enable a comprehensive analysis of NMJs. I focus particularly on the spatial organization of cell components, the structure of the muscle membrane, as well as the multi-scale analysis of the entire NMJ. For this purpose, I developed mathematical descriptions of central structures of NMJs and analysis tools for their relevant properties. On the basis of image data from modern microscopy techniques, I analyzed the structure of NMJs using graphical, topological, and statistical approaches.

In particular, the following contributions emerged from my research.

Visual Support for Segmentation Proofreading and Correction

Since classical segmentations (and methods) turned out to be inappropriate for our type of objects of interest, I supported our collaborators from Karlsruhe Institute of Technology in developing more intuitive segmentation approaches [7]. In [8] we introduced a two-dimensional abstraction of complex NMJ volume data and demonstrated its benefit for proofreading and correction processes of NMJ segmentations. Moreover, we proposed an automatic quality assessment of the resulting segmentation [9].

Data Set Creation

I developed data structures for the representation of membrane and fold information, both of which are crucial structural parts of NMJs, that also took into account different relations among the entities. Starting with the microscopic image data and initial segmentations, I developed tools for extracting and structuring the necessary information. As an example, I described my method for the extraction of junctional folds in [10]. As part of this endeavor, I have derived suitable novel descriptors from the extracted data objects. Hence, I created six data sets permitting the quantitative analysis of the fold and muscle membrane properties of NMJs for the first time.

COMPARATIVE ANALYSIS FOR CLASSIFIABLE DATA

The research process from the biologist collaborators is largely based on exploratory data analysis and hypothesis-testing. To this end, domain experts demanded quick and easy access to the data, to directly check different hypotheses in preparation for further studies. Moreover, this also permitted them to come up with more concrete questions. In [11] we introduced an interface I developed to simplify the comparison of subgroups of our data sets. Said interface combines feature selection and data representation. It thus communicates different kinds of information in a compact and visually clear manner. In addition, I use a similarity measure for comparing of distributions, that reduces the amount of potentially interesting subgroups that need to be inspected.

ABSTRACT REPRESENTATION OF TISSUE SAMPLES FROM NMJ

The abstract representation of the muscle cell membrane we introduced in [8] serves more purposes besides its support in proofreading segmentations: as a structure-preserving surface map of a central component characterizing an NMJ, it yields navigational support within the original volume data. It also provides a basis for structural analysis of other cellular components. The surface map and its interactive coupling with the original spatial data yields a template for biologists to organize complex three-dimensional data in a clear and intuitive way.

1.4 THE STRUCTURE OF THE THESIS

A good familiarity with the tissue structure at the NMJ is crucial for understanding the image data our work is based on. Hence, Chapter 2 gives an introduction on the biological background of NMJs. It explains the composition of muscle and nervous tissue and outlines the process of synaptic signal transmission. Furthermore, the chapter gives information about the image acquisition using electron microscopy and about their segmentation process.

In Chapter 3 we summarize existing work from scientific visualization that is related to the topics covered in this thesis. We refer to contributions from the area of segmentation proofreading, boundary segmentation and its quality control, segmentation editing, and 3D reconstructions from slice-based segmentations. For the analysis of descriptors of segmented biological image data we discuss existing work in the field of parameter space exploration.

We defined a data model for components of an NMJ, which we describe in Chapter 4. Said model is used in this work to structure the data for further processing and for analyzing NMJs.

Chapter 5 presents an algorithm for the decomposition of NMJ data into smaller entities. We introduce an interface for this algorithm that provides visual support for parameter tuning.

The segmentation of NMJ is a very challenging task due to the extremely intricate character of the objects of interest. Often, this is further hampered by the poor contrast of images obtained by electron microscopy. In Chapter 6 we introduce a framework with visual error control for image segmentation tasks. By means of this framework, we are able to obtain high-quality segmentations of NMJs.

The resulting segmentations may be structured according to the novel data model introduced in Chapter 4. Following this, Chapter 7 shows how to explore the parameter space of the resulting data in a structured manner. This exploration permits an analysis of the properties of NMJ components for the first time.

Finally, Chapter 8 concludes this thesis with a summary of the project's objectives, the results of this thesis, and an overview of ongoing work as well as numerous potential future tasks.

2 Background

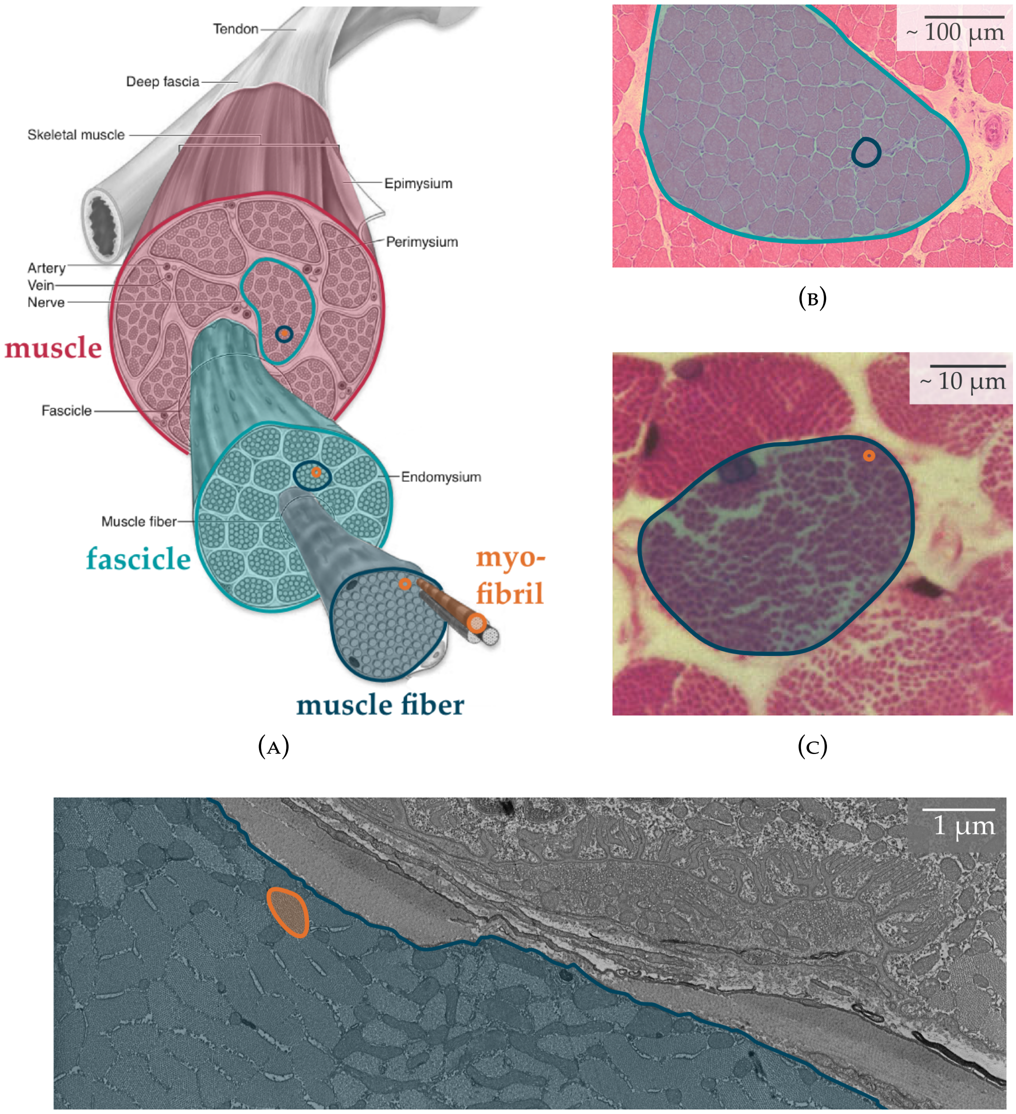
The data our project is based on comprises electron micrographs of muscle and nervous tissue, and their segmentations. This chapter gives a brief overview of the biological structure of muscle tissue and nerve cells from a visualization expert's point of view. A profound acquaintance with the schematic structure of these complexes and their interaction is crucial for understanding, interpreting and analyzing their real occurrences in images recorded by electron microscopy. The structures we see in these records are far more complex than what we usually see depicted in schematic descriptions. We will detail the acquisition of the image data, including a description of the sample tissue and its preparation. Furthermore, we will address image segmentation methods and their applicability to our data.

2.1 Biological Background

In this section we will introduce all cellular and structural components that are crucial for understanding the data of our project. We detail their embedding, interaction, and their functionality in the body.

2.1.1 Muscle Cell

Myocytes (muscle cells) are cells that form the muscle tissue. Skeletal muscles, cardiac muscle and smooth muscles are the three major muscle types. In the following description of muscle tissue we will refer to the skeletal muscle. Skeletal muscles are responsible for body movement. They are a form of striated muscle tissue. The stripes originate from repeatedly arranged units of *actin* and *myosin*. Bundles of these two proteins (myofibrils) shift into each other which causes the muscle to contract. In contrast to the other muscle tissue types, skeletal muscle cells are referred to as muscle fibers. These myocytes are very long and cylindrical like tubes and usually have more than one cell nucleus. They are laced up in fascicles. These are bundles



(A) (B) (C) (D)

Figure 2.1: Muscle hierarchy with microscopy examples. (A) The hierarchical structure of a skeletal muscle [12]. (B) and (C) Light micrographs show the muscle fiber (dark blue) in tissue cross sections in different levels of detail [13], [14]. (D) Cross-section of a muscle fiber as it appears in electron microscopic images.

surrounded by perimysium, which is a connective tissue. These bundles form the muscle tissue. This hierarchical structure of a muscle is shown in Figure 2.1a. This bundle structure can be well recognized in cross sections of muscle tissue recorded using light microscopy (Figures 2.1b and 2.1c).

Our focus lies on the muscle fibers, which we also refer to as muscle cells or myocytes. The cell membrane of a muscle fiber is called sarcolemma. It has a thickness of about 10 nm.

A myoblast is a progenitor cell of a skeletal muscle cell in the embryo. During embryonic development, several myoblasts fuse together to build a muscle fiber. Thus, each myoblast contributes a cell nucleus to a muscle fiber. These nuclei are located directly under the surface of the fiber, edged away from the cell inward where the myofibrils are located. The length of a fiber is up to several centimeters. Usually a single fiber reaches over the complete length of a muscle, for example from near one's elbow almost up to the wrist.

2.1.2 Neurons and Co.

In this section we give a brief introduction to cells that build the nervous system and their intercellular interactions.

Neuron

A neuron is a cell in the nervous system that is specialized for signal transmission and processing. It usually consists of a cell body (*soma*) in which its cell nucleus is located, *dendrites* and an *axon*. The dendrites are branching thin structures arising from the cell body. They are the neuron's "antennas" and receive information from other cells. The axon also arises from the cell body and propagates incoming signals further to other cells. In the human body it bridges distances between 1 µm and more than 1 m. At its end it may branch (multiple times) and forms the *axon terminal* or (pre-)synaptic bouton. These boutons interact with other cells, e.g. muscle cells or the dendrites of other neurons, and pass on information. Our focus will lie on these boutons.

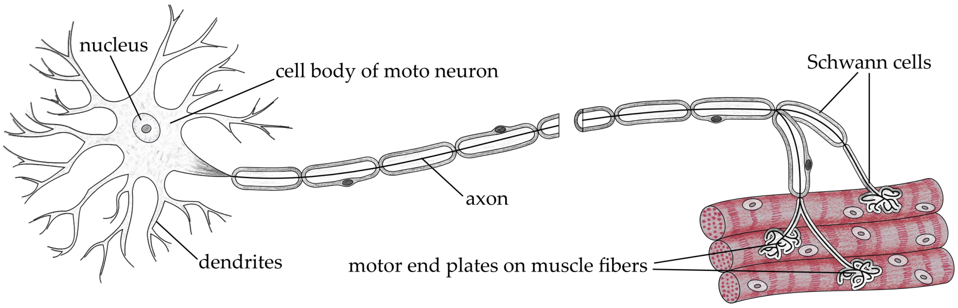


Figure 2.2: Schematic illustration of a motor neuron and connected muscle fibers. Image based on [15].

Motor neuron

The somatic motor neuron is a specialized kind of neuron. Its soma is located at the spinal cord and its axon leads directly to muscle fibers. This means that, for example, a single axon reaches from the spinal cord all the way down to the foot to innervate the muscles in its sole that are responsible for moving the toes. A single neuron may enervate multiple muscle fibers, but a muscle fiber is always enervated by only a single neuron. Hence, at the end of the axon it may branch multiple times and the branches end up in synaptic boutons, each connected to a muscle fiber. The synaptic bouton itself also branches multiple times. These branches cover an area on the surface of a muscle fiber. A more detailed description of the junction will be given in Section 2.1.4. Figure 2.2 illustrates the structure of a motor neuron.

Neurotransmission

A *synapse* is an interaction area between two neurons, or between a neuron and another cell. At these junctions electrical or chemical signals are transmitted. A typical chemical synapse is the junction of an axon terminal of a neuron and a dendrite of another neuron. In the axon terminal of the presynaptic neuron the neurotransmitter acetylcholine is stored in synaptic vesicles. This transmitter is released into the synaptic cleft, a gap with a width of about 20 nm between the two involved cells, when the presynaptic neuron is excited by an action potential. In the cell membrane of the postsynaptic cell receptors are located. They bind the acetylcholine molecules that were released into the cleft, which causes the excitation of the signal receiving cell.

2.1.3 GLIA

Glia cells (or just glia) form together with neurons the nervous system. They provide support and sheath for neurons and ensure electric isolation. Glia also influences the creation of synaptic connections [16, 17]. Only in recent years, glia was found to have own effects in neurotransmission itself [18].

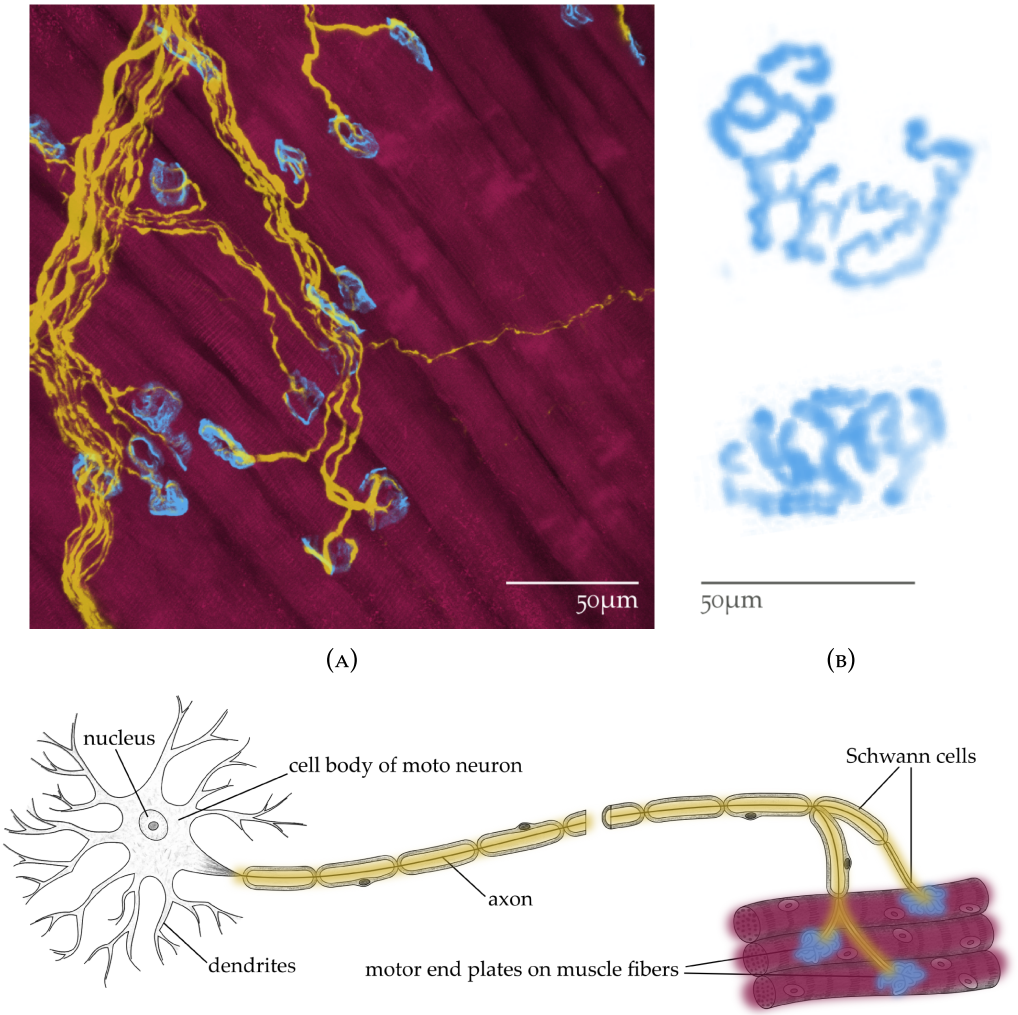
Schwann cells are a certain type of glia located in the peripheral nervous system, which is the nervous system outside the brain and spinal cord. They wrap around the axon of neuron building the so-called myelin sheath (see Figure 2.2) and cover the branches of the synaptic bouton.

2.1.4 NEUROMUSCULAR JUNCTION

An neuromuscular junction (NMJ) is a specialized synapse. It is the interface between a motor neuron and a muscle fiber. NMJs can be found in the middle of a muscle fiber's long side. A synaptic bouton of the neuron has a branching structure that covers an area of about 50 µm in diameter. "Fingerprints" of the branches of such boutons are shown in Figure 2.3b. The counterpart to the pre-synaptic boutons of the neuron, that lies beyond the synaptic cleft, is the motor end plate. This is the area on the surface of the muscle fiber that is covered by the bouton. At the motor end plate, nicotinic acetylcholine receptors (AChRs) are located in the sarcolemma of the muscle fiber. The binding of acetylcholine causes local depolarization of the motor end plate which spreads across the muscle cell's surface and causes the muscle to contract. Thus, this is the region where the neuron interacts with the muscle fiber by transmitting nerve impulses.

In Figure 2.2 depicting the illustration of a motor neuron we have already seen three NMJs at a coarse scale. In the following we will focus on this area in much more detail.

Figure 2.4 shows two samples of illustrations of NMJ as they are usually occur in textbooks. They are generally kept simple and focus on explaining the functionality of the NMJ. Even though these two examples are among the more detailed depictions that can be found, they do not show the true complexity of the NMJ's morphology.



(A)

(B)

(C)

FIGURE 2.3: Micrographs of neuromuscular junctions. (A) This micrograph [19] shows motor neurons each touching a muscle fiber (red). The axons of the neurons are dyed yellow. The postsynaptic parts of the NMJ (motor end plates) are labeled in blue via staining of acetylcholine receptors residing in the sarcolemma. This sample originates from a juvenile animal, thus, its scale is smaller and the synaptic boutons have not yet developed their full branching complexity. (B) This micrograph [6], however, shows motor end plates of an adult mouse. (C) Corresponding structures are colored accordingly in an schematic illustration (based on [15]).

(A) (B)

FIGURE 2.4: Schematic illustrations of neuromuscular junctions. Muscle fiber with motor end plate (synapse). The muscle membrane (sarcolemma) at the motor end plate forms *junctional folds*. Image sources: (A) [20]. (B) [21].

JUNCTIONAL FOLDS

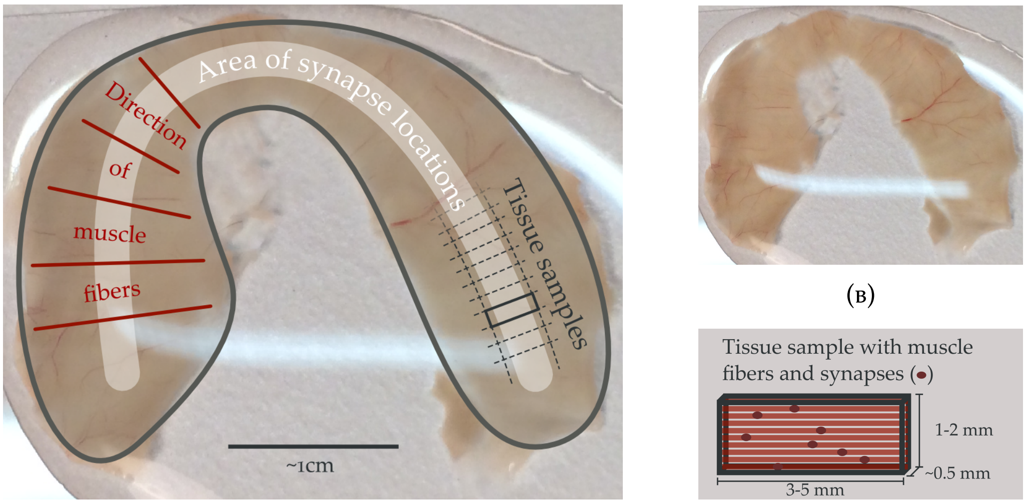
At the NMJ the sarcolemma features characteristic folds, the junctional folds. Both images in Figure 2.4 show them in a simplified manner. They suggest that these folds are like furrows in a field: regular in shape, straight invaginations into the muscle tissue. However, electron microscopic records show us that their real shape is much more complicated. In the micrographs of cross sections, perpendicular to the running direction of the muscle fibers, it can be observed that folds may have branches and sometimes a remarkable bending behavior in all spatial directions. Judging from these features, it can be derived that the sarcolemma describes a surface of complex shape in three-dimensional space.

2.2 IMAGE ACQUISITION

We work on stacks of digital images that depict a particular volume of muscle tissue. In this section we will describe the acquisition process of NMJs that leads from the tissue sample to the resulting image stacks.

2.2.1 TISSUE SAMPLE

The chest cavity and the abdominal cavity are separated by the *thoracic diaphragm*. The diaphragm is a very flat muscle, which is responsible for breathing movement. When contracted, the lungs get expanded and air streams into them. It belongs to the group of skeletal muscles. The diaphragm is a very thin and flat muscle with a horseshoe-like shape. See Figure 2.5b for an example of an extracted diaphragm of



(A)

(B)

(C)

Figure 2.5: Tissue samples from a diaphragm. (A) The running direction of the muscle fibers and the ribbon-type area where their synapses are located in a diaphragm. (B) The diaphragm of a mouse [19]. (C) Sample volume for the microscopy process, corresponding to a tissue samples in (A).

a mouse. The muscle fibers lie orthogonally to its medial axis. The synapses of the fibers, the locations where the fibers are innervated, lie in the middle of the fibers near the diaphragm's medial axis. These properties are illustrated in Figure 2.5a.

In our project, the tissue samples originate from mice. The diaphragm of a mouse has ca. 2.5 cm in diameter and is very thin with its thickness of 0.5 mm: Only about 10 muscle fibers contribute to its height.

Since biologists are interested in studying the influence of the tissue morphology on the proper function of signal propagation, pathological samples are examined. Therefore, tissue samples are taken from dystrophic mice that carry a *Dmd*^{mdx} mutation. In the 1980s Bulfield et al. [22] described a spontaneous mutation of the Dmd gene in one of their laboratory mouse strains. Since then, this mutation serves as an animal model of the severe disease *Duchenne muscular dystrophy*. This severe genetic disorder leads to progressive muscle degeneration and early death. It is caused by mutation of the gene *Dmd*, a gene that provides instructions for making the protein called dystrophin. A lack of dystrophin causes membrane destabilization and the activation of multiple pathophysiological processes [23]. We will refer to elements that result from these kind of samples adjectivally as *X chromosome-linked muscular dystrophy (mdx)* in the following.

To investigate such pathological alterations of tissue, however, it needs to be compared with corresponding healthy structures. In genetics the term *wild-type (wt)* refers to the typical form of an organism (or gene) as it occurs in nature. In general this means a healthy organism. In the following, elements that originate from such a sample are attributed with wild-type or its abbreviation *wt*.

Please note, that the six data sets that are presented in this work originate from only two individual animals.

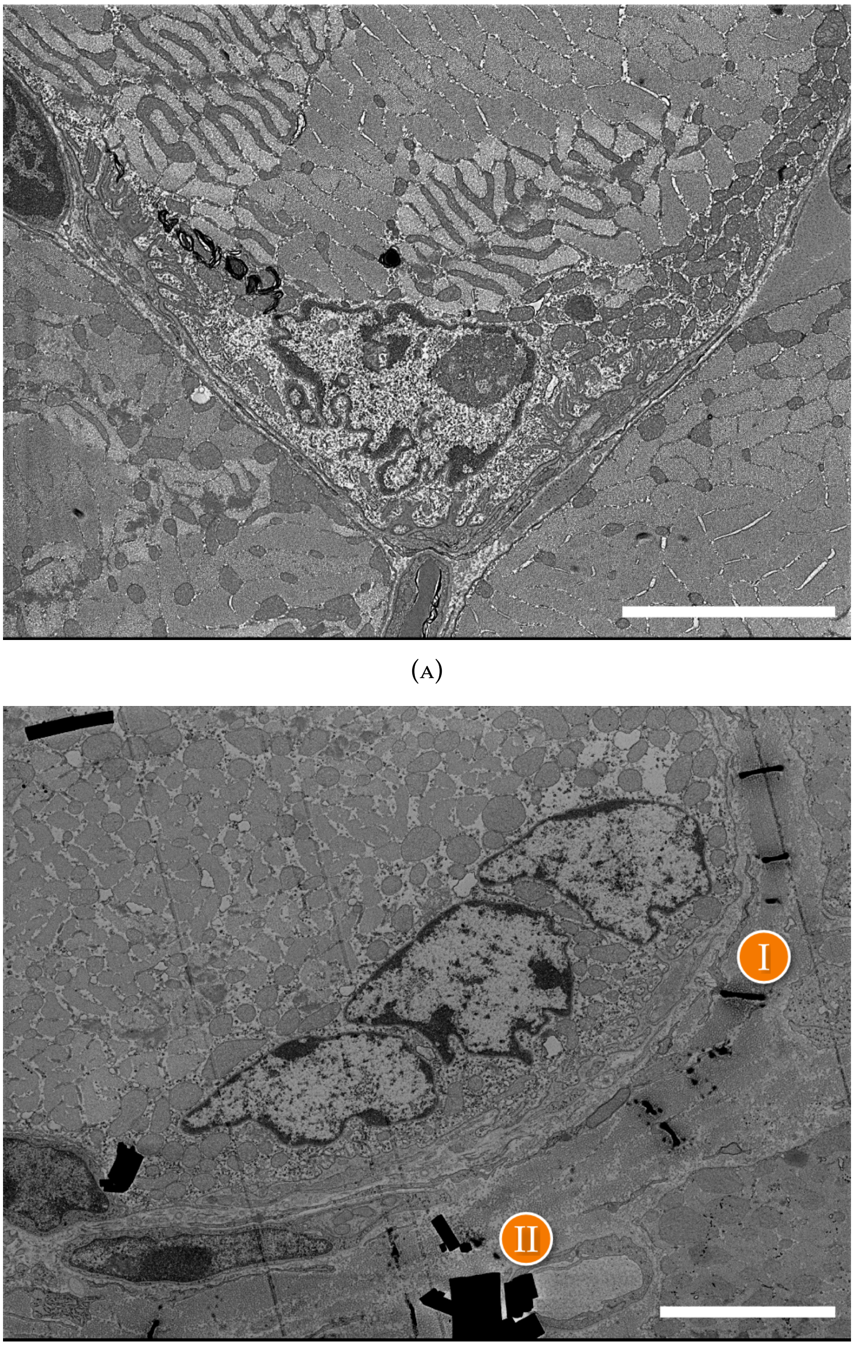
2.2.2 Specimen Preparation

The preparation of specimens for the electron microscopy process differs enormously from material to material. Dry specimen usually do not need to be prepared for the electron microscopy process, whereas tissue samples do. They need to be fixed and dehydrated. This process kills the tissue and can also change the appearance of the specimen. The choice of a suitable preparation method and its correct execution is also a critical point in the process of obtaining a high quality record.

In the first step of the preparation process the ribbon in the middle of the diaphragm, where the synapses are located, is extracted. This ribbon of tissue is then treated with heavy metal. The atoms of the metal bind to the tissue, enhance the contrast of the sample in the electron microscopy records. After this staining procedure the ribbon is cut into small pieces of 1 to 2 mm width (see Figures 2.5b and 2.5c).

Each small tissue piece is then embedded in a plastic block. They are fixed using synthetic resin or HM20 (which is another kind of resin). After the fixation the sample is cut into a cuboid with a width of about 1 to 3 mm and a length of 3 to 5 mm. Slices with a thickness of 70 to 100 nm can now be cut from the smaller side of this block with a diamond knife.

For the microscopy process, such ultra-thin slices are deposited on object slides. Using transmission electron microscopy (TEM) the electrons need to *pass through* the sample, so these object slides are small plates (diameter 3 mm) with a slot or a grid structure on which the sample is placed. That is why they are referred to as TEM grids. In the case of scanning electron microscopy (SEM), the electrons are *reflected* by the sample and the resulting secondary emission is captured by a detector. Thus, the detected electrons neither transmit the sample nor its support. For this reason, there is a much larger choice for the support for scanning electron microscopes (SEMs) as for transmission electron microscopes (TEMs). In our case,



(A)

(B)

Figure 2.6: Scanning electron microscopy images of NMJs taken from two of our image stacks. They are typical representatives of the image data we work with in this project. (A) is a slice from a wt sample, (B) originates from an mdx sample. The scale bars represent lengths of 5 µm.

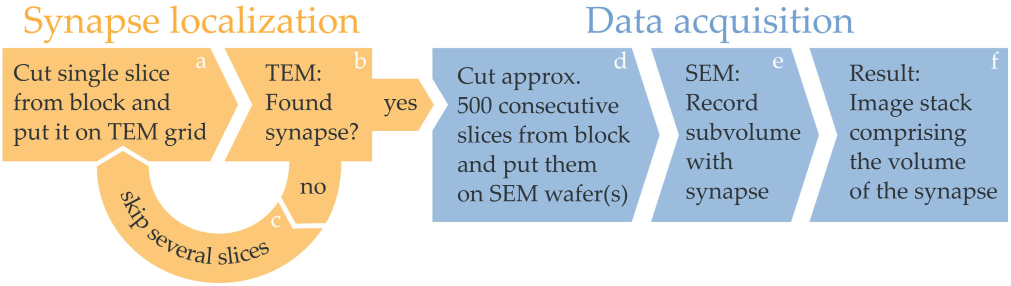


FIGURE 2.7: Acquisition workflow: The sample has to be manually searched for a synapse using transmission electron microscopy (TEM). Once a synapse has been found, its surrounding subvolume is recorded using scanning electron microscopy (SEM).

silicon wafers with a diameter of 2 cm were used. They provide space for about 100 consecutive sample slices. If necessary, the slices are now stained once again.

Inaccuracies during the preparation procedure may lead to artifacts in the resulting images. Black worm-like flaws (see Figure 2.6b, I for example) in micrographs originate from creases in the tissue slice when it was deposited on the sample support. Crystal-like artifacts may occur if the slices have been contrasted (see Figure 2.6b, II).

2.2.3 ACQUISITION OF AN NMJ

The tissue samples comprise volumes of 3 to 5 mm × 1 to 2 mm × 0.5 mm. An NMJ, however, covers an area of only about 50 µm × 100 µm, while having a depth of only a few micrometers. In other words, the NMJs are “hidden somewhere” in the sample. Thus, an NMJ first needs to be found before the subvolume in which it is located can be recorded with a sufficient scale.

The process of searching the synapse and its acquisition is illustrated in Figure 2.7 (a-f): An ultra-thin, 70 to 100 nm thick slice is cut from narrowest side of the tissue sample embedded in a small block of synthetic resin (a). This single slice is examined by TEM, because the employed TEM (EM900, Zeiss Microscopy, Germany) permitted a more comfortable and quick change of the magnification level and image section then the available SEM did. The microscope operator scans the slice for traces of synapses such as the characteristic folds of the sarcolemma (b). If the slice does not indicate a synapse, another slice is extracted from the sample block in a distance of approximately 2 to 5 µm to the previous one (c) and examined (a). This process is repeated until a synapse is found in the slice.

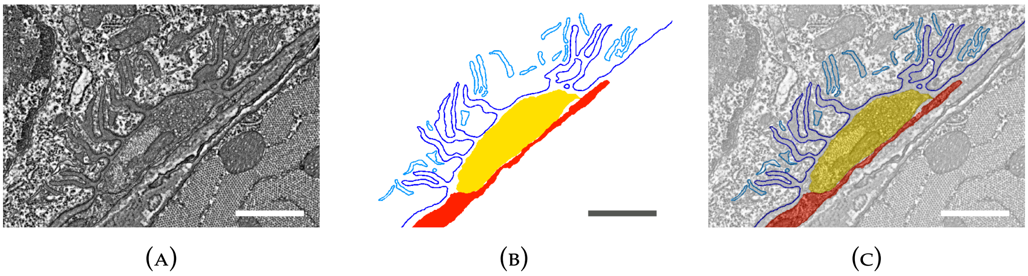


Figure 2.8: The labeling of neuron (yellow), glia (red) and sarcolemma (blue and light blue). Image detail of an NMJ. The scale bar width equals 1 µm. (a) EM image. (b) Object labels. (c) Overlay.

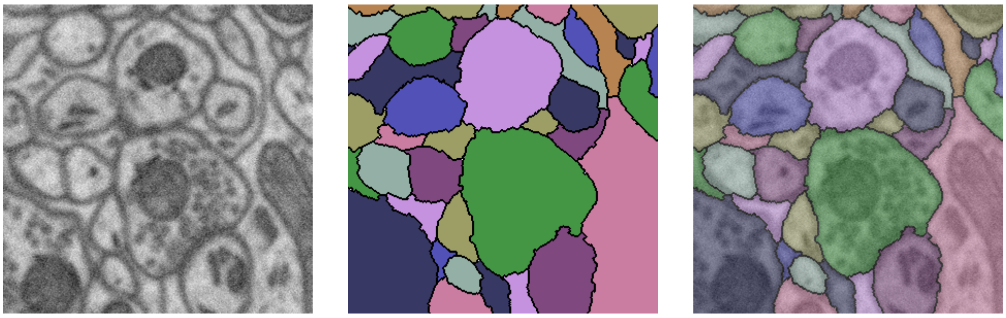
Once there is clear evidence for the existence of a synapse, the respective location is noted. Now about 500 consecutive slices are cut from the sample and (in groups of ca. 100) deposited on silicon wafers (d). The image section in which the synapse was found is now recorded with a SEM (Ultra 55, Zeiss Microscopy, Germany) for each of these slices (e). The result of the acquisition process is a stack of grayscale images comprising the volume of a part of an NMJ (f). Partial, because a part of the NMJ might already be cut away during the search procedure and only the subsequent slices are recorded.

In the meantime, our project partners are capable of recording larger volumes with SEM. Thus, prior localization of synapses is not necessary any more and image stacks comprising complete NMJs can be obtained directly now.

Figure 2.6 shows two typical representatives of the micrographs on which the following work is based on.

2.3 Object extraction

The human brain is extremely good at interpreting images and identify objects in them. We can tell which areas of an image (or pixels of a digital image) belong to a cow and which belong to the field it stands on, for example. Microscopic records of cell tissue are also images that the human brain can interpret, albeit with a certain amount of training. Looking at theses images, biologists can distinguish different objects such as cell organelles, e.g. mitochondria or nuclei, or cell membranes. However, for the computational processing of these different cell components and their properties, the location and shape of each of them have to be given explicitly.



(A) (B) (C)

Figure 2.9: Example of segmentation from Liu et al. [24]. (A) Original EM image. (B) Segmentation labels. (C) Overlay of raw image and segmentation. The labels match the cells.

In practice, all pixels that belong to a certain object are tagged with the same label. Usually, this pixel-label-assignment itself is represented by a (color) image, corresponding in size with the original image. Figure 2.8 shows the labeled objects (glia, neuron, sarcolemma) in NMJ records. All pixels that belong to the neuron cell, for example, are colored yellow in this figure.

The objects of interest can be extracted using different techniques that require varying amounts of (manual) intervention. First, the labeling of objects can be performed manually. Usually, this is done by virtually "painting" over the original TEM records and thus highlighting cohesive structures. The manual approach is very tedious and slow, but yields the best opportunities for quality control. Object labeling obtained by automatic image segmentation methods is known to be faster. However, such unsupervised approaches may yield insufficient results, often due to poor image input data. Semi-automatic segmentation techniques offer a good compromise. These hybrid methods combine the comfort of automatic approaches and the quality control and supervision of manual processing.

2.3.1 Image Segmentation

Image segmentation is a part of digital image processing. Segmentation refers to the division of an image into multiple regions based on specific homogeneity criteria, i.e., similar colors or textures are grouped together to segments. Figure 2.9b shows the segments as colored areas of the segmentation result of Figure 2.9a. The correspondence of original image and segmentation is shown in Figure 2.9c.

Khan [25] identifies three categories of segmentation methods: Pixel-based, region-based, and boundary-based. The first two result in defining segments by having each pixel assigned its respective segment affiliation. Boundary-based methods, however, yield closed curves. A segment is defined as a set of pixels that are surrounded by such a curve. The software Ilastik [26], for example, supports both supervised pixel- and region-based segmentation.

A critical issue of EM images for the computational segmentation, however, is the low contrast of the resulting images and the lack of color, which also obviates the possibility to (chemically) label specific structures in the tissue samples in order to highlighting them in the records. In our case, particularly the appearance of the sarcolemma turned out to suffer from this issue. Thus, completely unsupervised segmentation methods were not possible due to the complexity of the tissue structure (compared to much more simpler samples such as shown in Figure 2.9).

2.3.2 Segmentation of NMJs

For the analysis of relations between the cell organelles the segmentation of the microscopy records is needed. The different cell components that we want to examine have to exist as distinct objects, though. For our research question the “objects” glia, neuron, and sarcolemma are relevant.

Segmentations in this project were created by a biologist using *Amira* [27], a software for visualizing, manipulating, and exploring data from different two- and three-dimensional image acquisition devices. The unmodified EM images were used as input and the objects of interest were “carved out” by iteratively placing seeds manually and perform region growing. The resulting segmentation is region-based and exists as object labels, as shown in Figure 2.8b.

Even this computer-aided approach, however good it may be for areal shapes such as neuron or glia, is very cumbersome for narrow and elongated structures such as the sarcolemma.

3 Related Work

This chapter gives an overview of existing work that is related to the different topics covered in this thesis.

3.1 Proofreading of Segmentations

Visualization has seen increased interest over the last years in the proofreading of (semi-)automatic segmentations. For many large-scale applications of microscopy data, no perfect segmentation method exists and the data have to be manually corrected. Approaches addressing this problem often combine abstract representations of the segmentation to support the proofreading process and convenient correction techniques. A prominent example is the reconstruction of neural connections in connectomics research from large volumetric electron microscopy data. In Chapter 6 we introduce our contribution to the complex of proofreading segmentations.

Close to our approach is NeuroLines by Al-Awami et al. [28] who present an abstract representation of neuronal connectivity whose style resembles that of a subway map. While the authors focus on multiple neurons and their wiring, this thesis tackles the problem of segmentation proofreading and the closer analysis of the tissue structure at the location where a single neuron is connected to a single muscle fiber.

Finding correspondences between segments from adjacent slices is a central aspect of 3D reconstructions or the creation of abstract representations from segmentations. For example, Meyers et al. [29] present an approach for the determination of such correspondences for region-based segmentations. The information on corresponding segments is crucial for the reconstruction of connectomics data, where many segments tend to traverse the sample volume very densely. Helmstaedter et al. [30] use manually created skeletons as simplifications of corresponding segments. By comparing multiple skeletonizations of the same segmented object they verify the correctness of the segment correspondence and the underlying segmentation. Sicat et al. [31] obtain a three-dimensional graph structure from connectomics data using skeletons of the segments. They use this abstracted visualization to support the navigation in the volume data for a segmentation proofreading process. Both systems rely on three-dimensional abstractions because their objects of interest are isometrically distributed in their sample volume.

Haehn et al. [32] compare three proofreading tools for connectomics data and provide a requirements analysis for collaborative proofreading software design. They focus on direct proofreading and correction of region-based segmentation by multiple non-expert users, whereas our approach automatically reduces the amount of data that has to be manually revised in advance.

In [33], Al-Awami et al. concentrate particularly on the management aspect of large-scale project segmentation and correction.

3.2 Boundary Segmentation and its Quality Control

Boundary segmentation is a classical problem in computer vision and a large amount of techniques exist to detect such edges in images [34, 35, 36]. If the edges are rather thin, classical approaches such as the Canny edge filter give good results. As the boundary structure that we are interested in is often several pixels wide, we employ a *Livewire* technique [37] on pre-filtered images [38] that follows the user's mouse motion and snaps to highest-quality boundaries in the image.

Support for unsupervised quality control of edge-based segmentations is rare [39]. The most common choice are supervised evaluations where data is compared to ground truth [40]. In Section 6.3 we introduce a combined unsupervised quality analysis and a visual abstraction of the segmentation for error detection in surface segmentation evaluation.

3.3 Segmentation Editing

A large part of the proofreading process involves the correction of segmentations. There exist several approaches for the correction of region-based segmentations [41]. Most of them support a manipulation of the data volume by correcting the segmentations slice by slice.

Knowles-Barley et al. [42] present manipulations that consist of splitting and merging of segments and they support a sketch-based editing of segments in 2D.

For the correction process, Heckel et al. [43] introduce an approach where the underlying image data is taken into account. Subsequent work by the same authors [44] describes a method that is independent of the image data.

Karimov et al. [45] directly edit the volumetric reconstructions of the segmentations for correction. Grimm et al. [46] compile requirements for the editing process of planar polylines.

3.4 3D-Reconstruction

Finding correspondences between segmentations from adjacent slices is a central aspect of obtaining 3D reconstructions from slice-based segmentations. For example, Meyer et al. [29] present a well-elaborated approach for finding correspondences between areal segmentation of adjacent slices. Wang and Tang [47] introduce a method for the reconstruction of 3D surfaces from planar curves.

3.5 Parameter Space Exploration

Parametric descriptions of segmented biological image data are closely-related to work in the fields of parameter space visualization/exploration and image data analysis software for biological questions which we will review in this section. We present our approach for parameter space exploration in Chapter 7.

In visualization there has been much effort to visualize multi-dimensional parameter spaces. Starting with static methods such as parallel coordinates [48] and scatterplot matrices, increasingly sophisticated algorithms have been designed using, for example, multivariate projection (PCA, multidimensional scaling [49], *Worlds within Worlds* [50] or *HyperSlice* [51]). Jayaraman and North [52] provide an overview of an entire multidimensional function space, treating all dimensions uniformly and using a radial layout. Other approaches use sampling of the parameter space by extracting, for example, linear structures as proposed by Guo et al. [53].

Additional support is provided using animation or animated interaction to guide the user on their tour through the parameter space. Interactive parameter exploration for addressing different tasks and scopes is used by Tweedie et al. [54], where linked histograms depict the relation between the parameter space and the outcomes. Furthermore, an approach by Spence and Tweedie [55] is applied to gain insight in relations between attributes of multivariate data. Wittenburg et al. [56] provide a combined visualization and interaction interface to select subsets of the data by attributes using parallel bargrams. Further examples in this area are the *grand tour* [57], *rolling the dice* [58], or the *TripAdvisor*$^{N\text{-}D}$ [59].

Recent work resulted in some novel visual frameworks for exploratory data analysis: *Small Multiples*, *Large Singles* by Elzen and van Wijk [60], *Explates* by Javed and Elmqvist [61], and *Upset* by Lex et al. [62].

Apart from the methods that aim at a pure depiction and analysis of the parameter space, there exists an increasing volume of literature that focuses on the influence of parameter settings on an underlying model. This domain can be divided into two groups: The first one examines the impact of parameters on the model and wants to understand how changing the settings affects the model. The second group supports parameter selection by result, i.e., the user is presented a number of outcomes and explores the parameter space based on the outcome.

Examples for methods in the first group are *HyperMoVal* [63] and subsequent work by Berger et al. [64]. A strongly application-driven work of parameter influence is given by *Vismon* [65]. Another application where parameter space analysis is important is image segmentation. Representatives of these area are *Tuner* [66] and *Paramorama* [67], a plugin for *CellProfiler* [68]. The latter supports the parameter selection by giving the user the possibility to judge the quality of different results in order to reduce the parameter space. Hence, it may also be considered to belong to the second group: these methods sample the parameter space and compute results of discrete parameter combinations, so the user can explore the parameter space by result. There are numerous applications such as design galleries [69] with the explicit goal of producing visual output, that follow this idea. Bruckner and Möller [70] also use clustering to structure the results.

In addition, handling parameters that determine the rendering process of rendered scenes is a classical application area of visualization. In this context, Ma [71] introduces a directed graph layout to depict visualizations and their underlying rendering parameters. A node represents a scene and an edge pointing away from this scene depicts a rendering parameter that is changed. The node its pointing to is the re-rendered scene using the new parameter value. By interacting with this graph, users are able to comprehend complex parameter spaces. Another approach to address this task is given by Jankun-Kelly and Ma [72]. The rendered result is embedded in the parameter space. They used a three-dimensional representation of the parameter space, in which a pair of rendering parameters can be chosen to be variable and all others are fixed at this particular time. Pretorius et al. [67] display image results in the style of an image gallery, preserving the hierarchy from the parameter selection.

Analysis software designed for biological image applications often concentrates on cellular data. In this domain, the set of features is usually well-understood [73]. Hence, it is possible to use a variety of tools based on statistics and machine learning methods are used for the analysis of feature data. For a comprehensive review of methods in this field, we refer the reader to [74].

4 DATA PROCESSING

The most characteristic property regarding a neuromuscular junction are the folds that the sarcolemma forms. For the analysis of their properties and their relation to other cellular structures a number of diverse pre-processing steps needs to be carried through. This chapter will introduce the data model we developed to represent the structure of a neuromuscular junction (NMJ). Moreover, it will detail the derivation of data properties and their transfer to fit our data model. We also discuss the challenging aspects of retaining the consistency of our resulting data sets.

4.1 DATA

The part of the project that is covered by this thesis relies on two different types of image sources: Six stacks of microscopic gray-scale images, each originating from a sample of a different NMJ, and six corresponding stacks with segmentation images. The acquisition process of the microscopy images is detailed in Section 2.2. On average, every tenth EM images has a corresponding label image, which comprises the segmentation of glia, neuron, and sarcolemma. Section 2.3 describes the creation of these segmentations.

Three of the samples are taken from healthy mice. We refer to it as *wt*, which is short for wild-type. The other three samples are from dystrophic mice that carry a Dmd^{mdx} mutation, which we refer to as *mdx*. Table 4.1 lists information on the microscopic image data sets, the number of images, and the respective acquired volume size.

4.1.1 VECTORIZATION OF SEGMENTATIONS

As detailed before in Section 2.3.2 the sarcolemma plays a special role for the segmentation. In the segmentations, the sarcolemma is represented by a set of pixels that were found to belong it. Because of the rather one-dimensional shape of this

sample id	sample type	number of micrographs	number of segmented slices	volume size [µm]			image size [px]	
				x	y	z	width	height
1	mdx	58 (69)	11	15.33	11.50	4.76	3072	2304
2		101 (108)	11	16.78	12.59	7.49		
3		178 (187)	19	24.16	18.12	13.02		
4	wt	126 (150)	13	19.92	14.94	10.43		
5		126 (150)	12	20.12	15.09	10.43		
6		136 (150)	12	16.33	12.25	10.43		

Table 4.1: Properties of the microscopic image stacks. Image numbers in brackets include flawed records.

segment, compared to the areal two-dimensional shapes of neuronal or glial segments, a representation of the sarcolemma by a curve yields much better conditions for a further (shape) analysis of this segment.

We converted the pixel-based representation of the sarcolemma into a polygonal curve using the open-source vector graphics editor Inkscape, whose bitmap tracing is based on the *potrace* algorithm [75]. Due to the pixel-based segmentation technique the sarcolemma is often several pixels thick, whereas it also is interrupted at other points. These gaps were closed manually to obtain a connected curve for the sarcolemma. Figure 4.1 shows a detail of the segmentation before (left) and after (right) its vectorization.

The resulting vector-based representations of the segmentations provided the data basis for the statistical analysis part of our project. However, this part will be covered in Chapter 7. While the sparse sampling of these representations is sufficient for the development of techniques to support statistical analysis, the availability of more high-quality segmentations are necessary for a comprehensive analysis of NMJs.

4.1.2 Boundary-based Segmentation for the Sarcolemma

The labeling of the sarcolemma is the bottleneck of the segmentation process. The cross-section of a muscle fiber has a diameter of roughly 50 µm and is very heterogeneous in texture. The microscopic images we use comprise a region of about 20 × 15 µm (see Table 4.1 for details), which implies that a single image can depict at most one seventh of the fiber's cross-section. Hence, region- and pixel-based methods do not perform well in segmenting single muscle fibers in electron microscopic images of this scale. For the acceleration of the segmentation process a more direct segmentation of the sarcolemma curve is necessary.

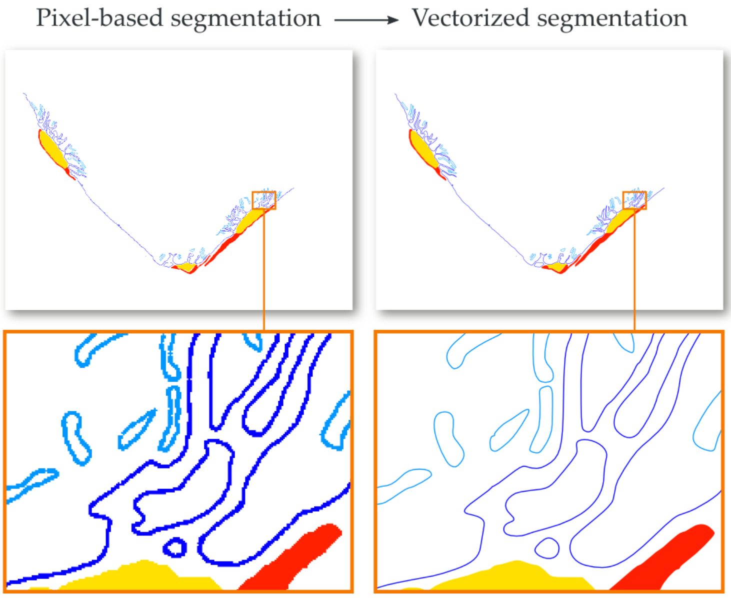


Figure 4.1: The segmentation of the slice shown in Figure 2.6: Sarcolemma in dark and light blue, glia in red, and neuron in yellow. Left: pixel-based segmentation, obtained semi-automatically by using the software AMIRA [27]. Right: Vectorization of pixel-based segmentation, vectorized with the free software *Inkscape*. Details of each of the two segmentations are shown in the bottom line.

Our records show a particular part of a muscle fiber and adjacent tissue. This implies that the cross-sectioned muscle fiber's membrane runs through the image without forming a closed boundary. In our work we use segmentations of this membrane, obtained by a modified variant of the semi-automatic boundary-based segmentation method *Livewire* [37]. The general idea of this approach is to identify a minimum cost path between two user-defined click points using a directed graph search algorithm. This Livewire method is originally designed to generate closed curves, however we modified it to permit tracing a section of a membrane, which usually does not lead to closed paths in our images for the reasons mentioned above [7]. Each resulting segment is represented by a polygonal chain (also referred to as *polyline*).

4.2 Geometric Definitions

Many of the data representation we deal with in this work are geometrical. Geometric data objects might be defined by a curve that represents their boundary or—if their shape permits it—by a curve itself.

For both cases we use polygonal curve representations, i.e. curves that are piece-wise linear and defined by a series of points.

In the following, we will introduce terms and symbols related to the polygonal curve representation that will be frequently used in this thesis.

4.2.1 LINE SEGMENTS

A **line segment s** is defined by its two *end points* **a** and **b**. It is the uniquely-defined straight line between **a** and **b**. Its parametric representation is given by

$$\mathbf{s}(\mathbf{a},\mathbf{b}) = \{\mathbf{a} + t\,(\mathbf{b} - \mathbf{a}) \mid t \in [0,1]\}. \tag{4.1}$$

The **length of a segment** is given by

$$|\mathbf{s}| = ||\mathbf{b} - \mathbf{a}||_2, \tag{4.2}$$

with $||\cdot||_2$ being the Euclidean norm.

In the following we will focus on particular sets of segments, namely polygons and polylines. Points that define the segments of polygons or polylines are called **vertices**.

4.2.2 POLYGONS

A **polygon** $\mathcal{P}_\circ$ is a piece-wise linear closed curve whose image is part of the two-dimensional Euclidean plane. A polygon is determined by a series of $n+1$ points $V := (\mathbf{v}_0, \mathbf{v}_1, \dots \mathbf{v}_n)$. Thus, in our case these vertices are all two dimensional ($\mathbf{v}_i \in \mathbb{R}^2$) and we refer to the coordinates of a vertex $\mathbf{v}_i$ with x_i and y_i.

Each pair of adjacent vertices of this series creates a **segment of the polygon**. The ith segment of a polygon is given by

$$\mathbf{s}_i = \begin{cases} \mathbf{s}(\mathbf{v}_i, \mathbf{v}_{i+1}), & \text{for } 0 < i < n \\ \mathbf{s}(\mathbf{v}_n, \mathbf{v}_0), & \text{for } i = n. \end{cases} \tag{4.3}$$

Note that, a polygon that is defined by $n+1$ vertices has $n+1$ segments.

Finally, we define a polygon as the union of all its segments:

$$\mathcal{P}_\circ = \bigcup_{i=0}^{n} \mathbf{s}_i. \tag{4.4}$$

4.2.3 Polylines

A **polyline** $\mathcal{P}$ (also *polygonal chain*) is a piece-wise linear open curve. Like a polygon it is defined by a series of $n+1$ points $\mathbf{v}_0, \mathbf{v}_1, \ldots \mathbf{v}_n$. In our case these vertices are situated on a plane and are two- or three-dimensional ($\mathbf{v}_i \in \mathbb{R}^2$ or $\in \mathbb{R}^3$, respectively). We refer to the coordinates of a vertex $\mathbf{v}_i$ with x_i and y_i for two dimensions and with x_i, y_i, and z_i for three ($_i, y_i, z_i \in \mathbb{R}$).

In contrast to a polygon, a polyline has no segment between its last and first vertex. The ith **segment of a polyline** is given by

$$\begin{aligned} \mathbf{s}_i &= \mathbf{s}(\mathbf{v}_i, \mathbf{v}_{i+1}) \\ &= \{\mathbf{v}_i + t\,(\mathbf{v}_{i+1} - \mathbf{v}_i) \mid t \in [0,1]\} \end{aligned} \tag{4.5}$$

for $0 < i < n$.

Thus, a polyline that is defined by $n+1$ vertices has n segments. As the polygon, a polyline is also the union of its segments:

$$\mathcal{P} = \bigcup_{i=0}^{n-1} \mathbf{s}_i. \tag{4.6}$$

4.2.4 Geometric Properties

The **length** $\mathcal{L}$ of a polyline equals the sum of the lengths of its segments:

$$\mathcal{L}(\mathcal{P}) = \sum_{i=0}^{n-1} |\mathbf{s}_i|. \tag{4.7}$$

Likewise, the **perimeter** (also *length*) $\mathcal{L}$ of a polygon also equals the sum of the lengths of its segments:

$$\mathcal{L}(\mathcal{P}_\circ) = \sum_{i=0}^{n} |\mathbf{s}_i|. \tag{4.8}$$

We will refer to the perimeter also with the term *length* for the sake of its equivalent in the case of a polyline.

The **partial length** $\mathcal{L}_k$ with $k \geq 0$ is defined as the sum of the lengths of the first k segments of a polygon or a polyline. Moreover, $k \leq n$ for polylines and $k \leq n+1$ for polygons. Hence, we have:

$$\mathcal{L}_k(\mathcal{P}) = \sum_{i=0}^{k-1} |\mathbf{s}_i|, \quad 0 \leq k \leq n \tag{4.9}$$

$$\mathcal{L}_k(\mathcal{P}_\circ) = \sum_{i=0}^{k-1} |\mathbf{s}_i|, \quad 0 \leq k \leq n+1 \tag{4.10}$$

The **area** $\mathcal{A}$ for non-self-intersecting (simple) polygons $\mathcal{P}_\circ$ is given by

$$\mathcal{A}(\mathcal{P}_\circ) = \frac{1}{2} \left| \sum_{i=0}^{n} (x_i \, y_{i+1} - x_{i+1} \, y_i) \right|, \tag{4.11}$$

where $n+1$ has to be evaluated to 0.

The following formula details the relation between polygons and polylines. Let $V = (\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_n)$ be a series of points. Then

$$\mathcal{P}(V) = \mathcal{P}_\circ(V) \setminus \{\mathbf{s}_n\}. \tag{4.12}$$

This allows the transformation of a polygon to its corresponding polyline and vice versa, supposed that the polyline is planar and has no self-intersections.

4.3 Data Model

An ideal initial segmentation results in isolating individual objects that refer to different cellular or sub-cellular items. Each neuron and glia cell is represented by a connected subspace in the three-dimensional space, i.e. by a solid 3D volume. The sarcolemma can be modeled by a two-dimensional surface embedded in a three-dimensional space. The junctional folds that are formed by the sarcolemma subdivide the sarcolemma in smaller parts.

For the organization of the different entities we are dealing with, we developed a hierarchical data model for the representation of a neuromuscular junction. The model is influenced by biological circumstances as well as by data-specific technical properties.

The entities in their hierarchical order from top to bottom are: The synapse, the slice, the membrane and the fold. The following sections will introduce and define each of them in detail.

4.3.1 Synapse Object

A *synapse object* is an abstract representation of the synapse that a microscopy image stack comprises. This object has a number of attributes that depend on the tissue sample and the data acquisition process.

The *identifier* tells to which data set the image stack belongs to. Typically this is an integer.

A synapse object comprises a stack of microscopic images of equal scale. The *scale attribute* numbers the width of the rectangular area that is depicted in such an image. Within our data sets this value lies between 15 to 25 µm.

Health is a textual attribute that indicates the health status of the animal from which the image data origins. In our data sets, the possible values are `wt` for wild-type animals or `mdx` for animals suffering from X chromosome-linked muscular dystrophy.

4.3.2 Slice Object

A *slice object* represents a slice of the data volume that is recorded by an microscopic image. The volume is cut into slices and each microscopic image of the image stack shows a slice. A slice object stores the attribute *index*, which refers to the position of the image within the stack. Moreover, the slice might contain image data of one or more images itself. Usually, the microscopic image and the label image are stored.

This entity is of no biological meaning, but results from the organization of the data in image stacks.

4.3.3 Membrane Description

The sarcolemma appears in cross-sections as a curve that usually traverses the microscopic image. This curve separates muscle tissue from the synaptic cleft, the intercellular space between the muscle cell and a neuron. Figure 4.2a shows these two regions and how the sarcolemma separates them. Note that each of both (muscle tissue in red and exterior of muscle tissue in grey) is not necessarily a connected region.

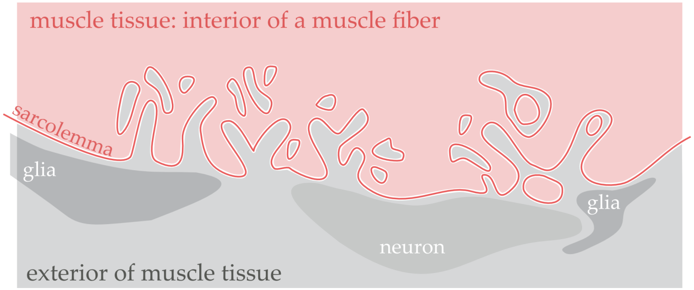
In direct proximity to where the membrane features the characteristic folds, the sarcolemma also appears as smaller closed curves, which result from evaginations of the sarcolemma in orthogonal direction to the sectional plane. Most of these regions that are enclosed by these curves lie within the muscle tissue. They are enclaves of the muscle fiber's exterior in the muscle tissue. Exclaves of the muscle tissue, which occur much less often, lie on the other side of the sarcolemma, surrounded by the exterior of the muscle cell. Figure 4.2b illustrates this situation. This covers most of the cases pertaining to the appearance of the sarcolemma in our data. We observed some additional more exotic constellations, such as enclaves of higher order, but for now, we do not consider them in our investigations.

At a synapse, vis-a-vis the muscle tissue, across the synaptic cleft, may lie glia tissue, a neuron, or nothing but more intercellular space. Figure 4.2c shows how the sarcolemma is subdivided and its parts are assigned both according to the adjacent cell tissue. Biologically the sarcolemma of an entire synapse object is one single object, i.e. one single surface in the three-dimensional space. However, for organizational reasons, a membrane object in this thesis refers to a single curve in a two-dimensional slice which is either open or closed. In the following we define the attributes of a membrane object.

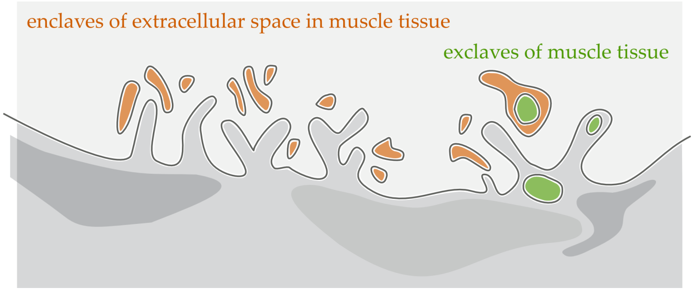
We denote the different kinds of appearance of the sarcolemma with the term *membrane type*. The membrane type of a curve may be *open* or *closed* as shown in Figure 4.2d. An open membrane may potentially feature folds, whereas closed membrane parts are not considered to have folds in our data model.

Another attribute of a membrane object is its *vertex data*. In practice we model the curves that represent the cross-sections of the sarcolemma by polygonal chains or polygons. This attribute stores the sequence of vertices that define the polygonal curve.

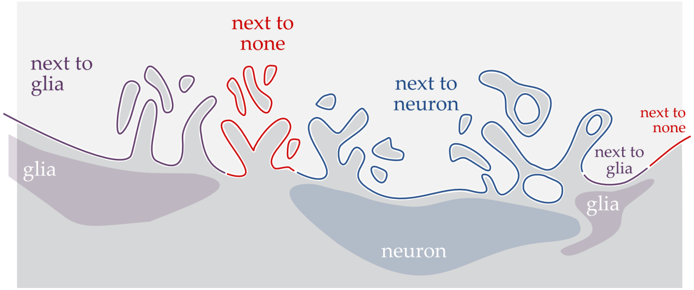
The attribute *nearest tissue* specifies which cell—if any—lies next to a membrane part. Possible values for this attribute are *neuron*, *glia*, or *none* (also called *interspace*).



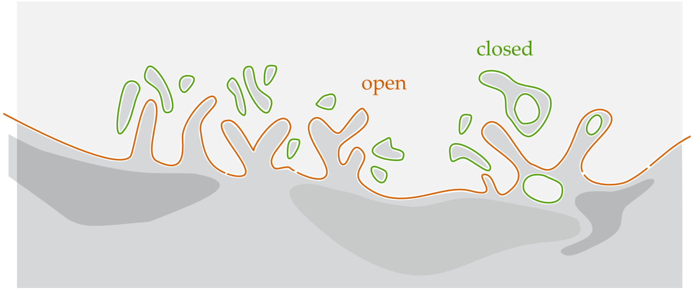
(A) Components of an NMJ



(B) Appearances of the sarcolemma



(C) Sarcolemma assigned by neighborhood



(D) Types of curves describing the sarcolemma

FIGURE 4.2: Schematic illustrations of the sarcolemma at a neuromuscular junction as it appears in microscopic images of volume cross-sections. Each figure shows the categorization of membrane components according to different aspects by means of different color coding.

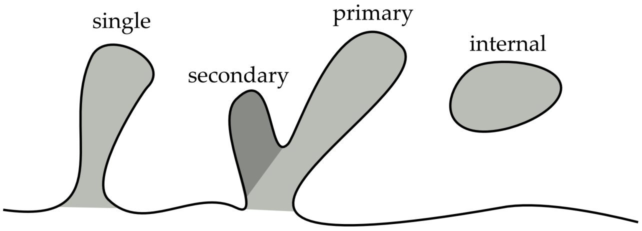


FIGURE 4.3: The four different types of folds that we differentiate in this thesis.

4.3.4 FOLD OBJECTS AND THEIR CLASSIFICATION

A fold is defined as a connected subchain of the polyline that represents the cross-section of the muscle cell membrane. Thus, a reference to the membrane object as well as the indices of the vertices of the membrane's polyline that determine the start and the end point of the fold define a range where the membrane forms a fold. The determination of the range boundaries from a given membrane is described in Chapter 5. For now, we assume that we are already given these ranges.

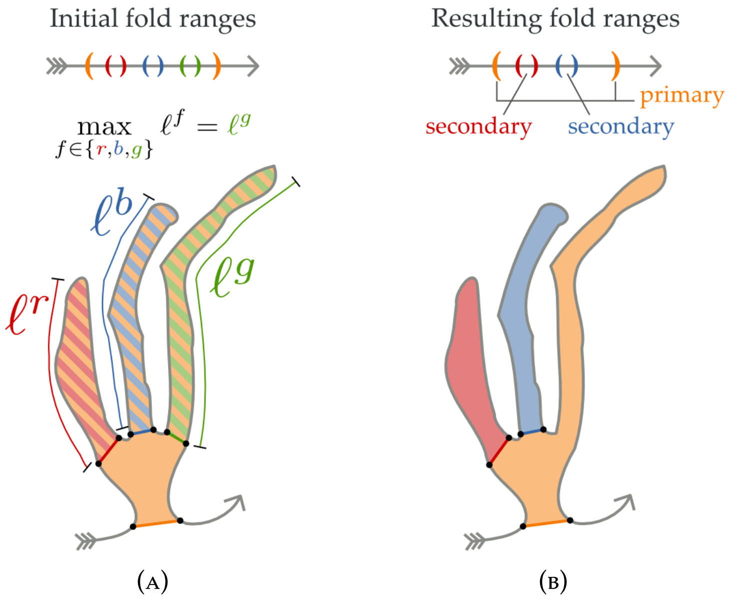
We differentiate between four types of folds: *single*, *primary*, *secondary*, *internal*. These types are illustrated in Figure 4.3 and described in detail in the following.

Single folds are the prototype of a fold. They are given by an interval on the membrane as simple invaginations of the extracellular space into the muscle tissue.

Occasionally, folds feature a branching behavior. Figure 4.4 illustrates the classification and treatment of such branching structures. If we regard the folds as ranges on the membrane, this is when one or more ranges lie within another range. We consider all folds that are defined the inner ranges to be **secondary folds**—except for the case of the fold of greatest length. The outer range forms the **primary** fold. Section 4.4.2 provides further details about a formal definition of the *length of a fold* and how it is determined. At this point here the intuitive comprehension of the fold length suffices.

Primary folds are defined by their start and ending index just as the definition of single folds. As a consequence, a primary fold shares parts of the membrane with its secondary folds. However, for further data extraction, it may be necessary to remove the inherent secondary folds (see Section 4.4).

A somewhat special case of folds are the **internal** folds. Internal folds are the most common folds in our data. An internal fold is an enclave of extracellular space in muscle tissue in a cross section. For this reason, it has no opening. A large amount of



(A) (B)

Figure 4.4: The classification and treatment of branching folds. (A) Initial situation of branching folds as they result from the fold detection process. The range of the orange fold contains three other folds (red, blue, green). Let ℓ be the length of each fold as depicted. (B) The fold of longest length (green) merges into the outer fold (orange) forming the *primary fold*. The other folds (red and blue) are *secondary folds*.

internal folds indicates the high complexity of the membrane's morphology which extends in all spatial dimensions. It shows that the membrane has not only furrows as, for example, corrugated iron, but rather convexities in direction of the main fold running direction.

Internal folds require a special treatment in our examinations. In some cases we have to exclude them from the data pool because they do not permit certain operations or provide particular features. For example, they are not directly suitable for an examination of the depth of folds, because they do not have an opening.

4.3.5 Hierarchical Data Model

In the preceding sections we introduced the entities of our data model. Each object has a number of attributes. Some of these attributes are determined by means of domain knowledge or store acquisition information, others are derived directly from the geometrical representations of the objects. Figure 4.5 roughly reflects this circumstance by placing attributes of the latter class below their objects.

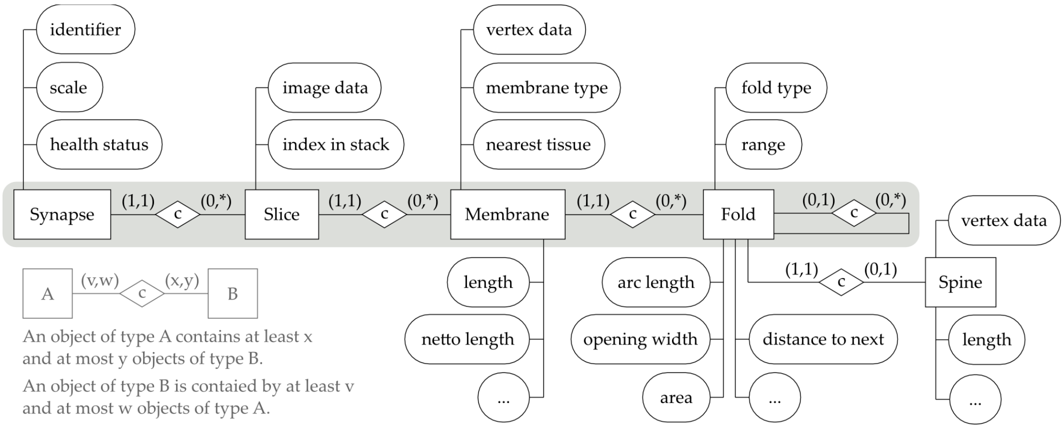


Figure 4.5: The entity relation model (ER model) of our data. The gray area highlights the hierarchy of the main entities synapse, slice, membrane, and fold. The cycle to the right of the fold entity models the relation between primary and secondary folds. Although the spine is modeled as an entity for implementation purposes, it is rather a property of a fold, and therefore not included in the main hierarchy. Attributes located below their entities are derived from attributes that are located above their respective entities. The attributes with dots indicate that their model can be expanded at these locations by additional attributes if desired.

Having defined all required attributes, we now describe the dependencies between these entities. Our NMJ data model is organized hierarchically. The hierarchy is depicted in Figure 4.5 from the synapse object on the left hand side to the fold object on the very right. A synapse comprises a number of ordered slices. Each of these slices contains several membrane objects (parts of the curve that model the sarcolemma, see Section 4.3.3). Each membrane object, again, may potentially comprise several folds. Open membranes are not restricted in the number of folds they can have. Closed membranes, however, are a special case: if they are enclaves of extracellular space in the muscle tissue, they are forming a fold in their entirety. We refer to folds of this type as *internal* folds. Finally, primary folds on their part at least one dependent fold.

4.4 DATA PROPERTIES

It is possible to define multiple measured properties to the different entities in our data model. Most of them are easy to compute because they are based on the length of a polyline, for example, or the area of a fold, which is basically the area of the

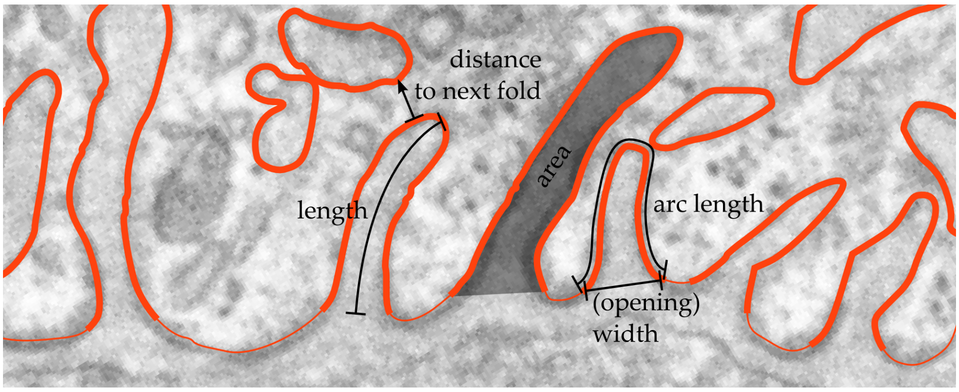


FIGURE 4.6: Fold features

polygon that describes the fold. Others require auxiliary constructions. Figure 4.6 shows our NMJ data model augmented by the measurable properties of its entities.

4.4.1 Direct Fold Properties

Let F be a single or secondary fold. A fold F is geometrically represented by the polyline $\mathcal{P}^F$. Let $\mathcal{P}_\circ^F$ be the corresponding polygon of polyline $\mathcal{P}^F$ according to Equation 4.12.

The following three properties $\texttt{\textbf{arclength}}^F$, $\texttt{\textbf{area}}^F$, and $\texttt{\textbf{width}}^F$ only depend on a solitary fold.

The *arc length* of a fold is given by

$$\texttt{\textbf{arclength}}^F := \mathcal{L}(\mathcal{P}^F). \tag{4.13}$$

It is the length of the polyline that geometrically represents the fold F.

The area that a fold covers is given by

$$\texttt{\textbf{area}}^F := \mathcal{A}(\mathcal{P}_\circ^F). \tag{4.14}$$

The *(opening) width* of a fold is defined as the width of the gap formed by a fold. It is given by the Euclidean distance between its two end points, $\mathbf{v}_0$ and $\mathbf{v}_n$, of $\mathcal{P}_F$:

$$\texttt{\textbf{width}}^F := \|\mathbf{v}_n - \mathbf{v}_0\|_2 = \mathcal{L}(\mathbf{s}_n^{\mathcal{P}_\circ^F}). \tag{4.15}$$

The *distance to the next fold* is also a property of a single fold F. However, its calculation requires taking into account all other folds of the same slice as F. Let ψ be the union of these folds (without F).

$$\mathtt{\mathbf{distance2next}}^F := \min_{\substack{\mathbf{v}^F \text{ of } \mathcal{P}^F \\ \mathbf{v} \text{ of } \bigcup_{G \in \psi} \mathcal{P}^G}} \|\mathbf{v}^F - \mathbf{v}\|_2 \tag{4.16}$$

Three of the four presented properties can also be determined for an internal fold F, namely $\mathtt{\mathbf{arclength}}^F$, $\mathtt{\mathbf{area}}^F$, and $\mathtt{\mathbf{distance2next}}^F$.

Special Treatment of Primary Folds

For primary folds, the calculation of their properties has to be modified. The secondary folds of a primary fold have to be taken into account when calculating its **`area`**and **`arclength`**. Let B be the set of secondary folds of a primary fold F. Thus, the *arc length* of a primary fold F is given by

$$\mathtt{\mathbf{arclength}}^F := \mathcal{L}(\mathcal{P}^F) + \sum_{\beta \in B} (\mathtt{\mathbf{width}}^\beta - \mathtt{\mathbf{arclength}}^\beta). \tag{4.17}$$

Moreover, the *area* of a primary fold F is given by

$$\mathtt{\mathbf{area}}^F := \mathcal{A}(\mathcal{P}_\circ^F) - \sum_{\beta \in B} \mathtt{\mathbf{area}}^\beta. \tag{4.18}$$

4.4.2 Length of Folds

The fifth fold property that we introduce is the *length of a fold* or—regarded from a biological point of view—rather the *depth of a fold*. The properties we have presented so far are directly derived from the polygonal representation of a fold. For the calculation of the length of a fold, however, we need to make use of auxiliary geometry. Note that this property does not hold for internal folds, since they do not possess an opening and thus do not provide something that could be considered as a depth.

Our auxiliary geometry is the fold's *spine* that we derive from the given fold geometry. The spine is a curve. We consider it to arise from the middle of the fold's opening, travelling down the fold, while following its shape and to end it at the fold's cap. The *length* of any non-internal fold F, eventually, is given by the length of its spine S^F:

$$\texttt{length}^F := \mathcal{L}(\mathcal{P}^{S^F}). \tag{4.19}$$

Spine Determination

The spine is a polyline and can be considered as a part of the medial axis of a fold shape. The medial axis of a shape is the union of points (inside the shape) that have more than one closest point on the shape's boundary. In two dimensions the medial axis of a shape can be approximated by means of a *Voronoi-diagram* of points sampled equidistantly from the shape's boundary.

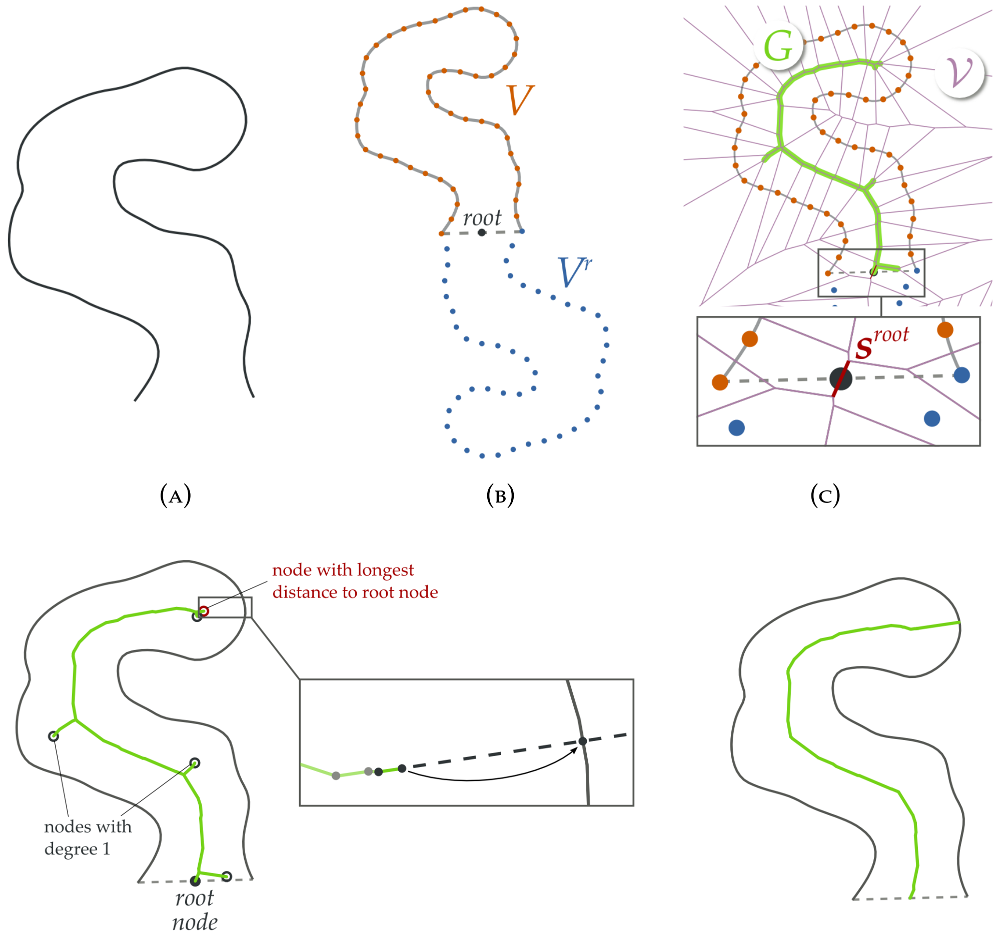
For the calculation of the spine we use a graph structure derived from a Voronoi-diagram that we call *skeleton graph* to approximate the medial axis of a fold. Using Dijkstra's algorithm for finding the shortest path in graphs we eventually extract the spine from the skeleton graph. We start with a definition of the required structures.

The **Voronoi diagram** partitions the Eucledian plane with n points S—the *Voronoi sites*—into n convex regions, the *Voronoi cells*, so that each cell contains exactly one site and all points in a cell are closer to this cell's site than to any other site (according to the Euclidean distance) [76]. The resulting Voronoi diagram $\mathcal{V}(S)$ consists of a set of line segments and half-lines (or lines, for special cases, as it is the case for collinear Voronoi sites), which separate the regions in the plane.

We define a **skeleton graph** $G(V, E, W)$ as the connected graph that results from $\mathcal{V}(S)$, with V is the set of vertices that consists of the start and end points of the voronoi-diagram's line segments, E is the set of pairwise connections between the vertices in V given by the segments of $\mathcal{V}(S)$, and W are the weights of the edges given by the length of the segments of $\mathcal{V}(S)$.

In the following we will detail the pipeline we developed in this project for determining the spine of a fold. Figure 4.7 illustrates the workflow. The pipeline consists of four major steps:

1. Preparation of point data for the creation of the Voronoi diagram $\mathcal{V}$
2. Reduction of $\mathcal{V}$
3. Extraction of the spine in the skeleton graph G via Dijkstra's algorithm
4. Localization of the last vertex of the spine



(A) (B) (C)

(D) (E) (F)

FIGURE 4.7: The determination process of a fold's spine. (A) Polyline of a fold. (B) Sampled vertices of fold and their point reflection. (C) Voronoi diagram and skeleton graph. (D) Finding the longest path. (E) Elongation of last segment. (E) Resulting spine.

For the spine determination of a fold F (Figure 4.7a) we sample its polyline $\mathcal{P}^F$ equidistantly and with sufficient density to preserve characteristic morphological features of our data. A stride length of the median of polyline's segment length yields satisfying results. Let $V = (\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_n)$ be the sampled data points.

Let $root = \frac{\mathbf{v}_0 + \mathbf{v}_n}{2}$ be the point be the center point of the opening of a fold, where the spine starts.

Let V^r be the set of sample points that result from a rotation of V by 180° about the point $root$. Let U (Figure 4.7b) be the union of both the original sample point set V

and its reflected version V^r:

$$U = V \cup V^r. \tag{4.20}$$

Including the reflected versions V^r of the sample points V ensures that there will be exactly one segment $\mathbf{s}^{root}$ of the resulting Voronoi diagram $\mathcal{V}(V \cup V^r)$ that intersects segment $\mathbf{s}_n$ of $\mathcal{P}_\circ^F$ in point $root$ (Figure 4.7c). The existence and uniqueness of such a segment allows us to define the following subset of $\mathcal{V}(U)$.

Let $\mathcal{V}^*(U)$ be a reduced version of $\mathcal{V}(U)$:

$$\mathcal{V}^*(U) = \{\, \mathbf{s} \in \mathcal{V}(U) \mid \mathbf{s} \text{ lies completely in } \mathcal{P}_\circ(V) \,\} \cup \mathbf{s}^{root/2} \tag{4.21}$$

where $\mathbf{s}$ are line segments of the Voronoi diagram and $\mathbf{s}^{root/2}$ is the half part of segment $\mathbf{s}^{root}$ which lies in $\mathcal{P}_\circ(V)$.

Let $G(V, E, W)$ be the skeleton graph resulting from $\mathcal{V}^*(U)$ (Figure 4.7d). By construction, $root$ is a node of this graph. For regular folds, there is a clearly-visible recognizable path in the skeleton graph that follows the main course of the fold. This path is the longest of the shortest paths that exist in the graph between the root node and any other node with degree[1] 1. For determining these shortest paths in the skeleton graph we use Dijkstra's algorithm [77]. The longest of these paths is the major part of the fold's spine.

Usually the medial axis of a shape does not extend to the border of the shape (except where the shape shows corners). Its approximation via the Voronoi diagram behaves similarly: the skeleton graph does not extend to the fold's cap. Thus, the last vertex of the spine is relocated in such a way that its end vertex lies on $\mathcal{P}(V)$ and the direction of the segments remains unchanged (Figure 4.7e).

4.4.3 MEMBRANE PROPERTIES

$\mathcal{P}^M$ is a polyline representation of the membrane.

The *arc length* (see Figure 4.8a for illustration) of the membrane is given by

$$\texttt{arclength}^M := \mathcal{L}(\mathcal{P}^M). \tag{4.22}$$

---

[1] The degree of a node is the number of edges that it is connected with

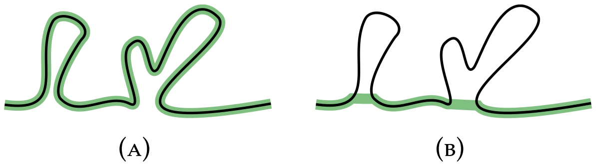

(A) (B)

FIGURE 4.8: Properties of membrane objects. (A) Length of a membrane object. (B) Netto length of a membrane object.

If we image that all the folds within a membrane are removed, then the *netto length* of the membrane is the distance the membrane travels while skipping all its folds. Figure 4.8b illustrates this property. Let Ω be the set of primary and single folds that depend on M. The netto length of a membrane M is then given by

$$\mathtt{\textbf{nettoarclength}}^M := \mathcal{L}(\mathcal{P}^M) + \sum_{F \in \Omega} (\mathtt{\textbf{width}}^F - \mathtt{\textbf{arclength}}^F). \tag{4.23}$$

4.5 DATA SETS OF NMJ COMPONENTS

After obtaining characteristic values from NMJ components, it is possible to create derived numerical data sets from them. These derived data sets may in turn be subject to a statistical analysis of properties of NMJ components. For example, we created a data set that consists of parts of membranes: each object in the data set refers to either a fold itself or the part of the membrane between two folds. In addition, every item of the data set stores its properties, such as *length* or *nearest tissue*. Data sets of this kind are used by our biologist collaborators in order to determine the percentage of membrane directly situated next to a neuron. Likewise, this sort of data set permits determining how many folds are within one micrometer length of the synaptic cleft.

Regardless of the individual characteristics of such derived data sets, they should be kept statically in general: operations such as removing a fold may have ramifications for the remaining data. However, if the dependencies between individual items are unknown, it is impossible to adjust the relations between them automatically.

Nevertheless, changes in the underlying data are required if data need to be corrected, for example: if we assume that a membrane has been segmented incorrectly at its beginning, it may require manual corrections. These corrections may result in a shortening of the membrane. Following our data model, we now would need to adjust the start and end indices of all folds that are part of the membrane. Otherwise, the data set runs the risk of becoming inconsistent.

In other words changes in the data may globally influence the data. Implementing a data model that supports changes hence implies that dependencies—both direct and indirect—between items need to be considered in order to preserve data consistency.

5 Visual Parameter Control for Fold Extraction

Parts of this chapter have been already published as a technical report [10] to serve as a reference for our subsequent publications.

Image segmentations are a common part of the analysis of image data. They are used to identify different objects, for example cell nuclei, mitochondria, or complete cells in microscopic images. But there might be features in the data, that cannot be detected by image segmentation approaches directly, because they are not characterized by their texture or boundaries, which are properties most segmentation techniques rely on, but of a morphological nature. This necessitates decomposing the results from image segmentation. In this chapter we will introduce an algorithm for the decomposition of segmented objects from segmentations of neuromuscular junctions. We face the well-known problem of finding the best parameters for an algorithm by presenting an interface for the algorithm that supports informed parameter tuning.

5.1 Introduction

In our application, biologists are interested in junctional folds and their properties. Junctional folds are parts of the muscle cell's membrane that burrow like fingers into the surrounding muscle tissue. In the image data, the membrane, is represented by a curve-like structure. The segmentation of this structure is given by a curve, which does not distinguish between individual folds. Hence, we have to decompose the curve into “fold” and “non-fold” parts to extract the folds in order to compute fold characteristics such as width, depth, or branching style. All folds on such a curve are oriented to its same side. The main course of the curve might also bend (on a larger scale). In the following, we assume the curves to be piece-wise linear. In Chapter 4 we stated that a fold is given by a reference to the membrane it belongs to, combined with the indices of its first and last vertex on the polyline that represents the membrane. In the following we will present a method for determining these indices.

Our algorithm uses three parameters. Commonly, parameters of algorithms are tweaked by users until their output data shows the desired result. Even if the users know theoretically which parameter to be responsible for which property of an algorithm it is often not clear what the effects of changing a single parameter are. This is even made more complicated in case the interactions/dependencies between parameters are not made clear. We visualize the functional principle of the algorithm and which influence each of these parameters exercises on the parts of the algorithm—and ultimately, on the result. Thus, the appropriate parameter values for the given data can be found very easily, because their effects pertaining to the algorithm are communicated clearly to users.

5.2 DEFINITIONS

We define a *fold* as a (connected) subset of a curve, satisfying two conditions:

1. The subchain contains a turning angle greater than a given threshold τ. A subchain that satisfies this condition is called a *minimal subset*.
2. The first and the last point of the subchain have a maximum Euclidean distance of 2δ and the arc length of the subchain is $\gg 2\delta$. A subchain that satisfies this condition is called a *maximal subset*.

Condition 1 specifies that we are looking for parts of the membrane that feature a tip and condition 2 requires the structures to have a small baseline, i.e., that the membrane has a close distance to itself at the fold's "foot". While the first condition determines whether a fold is present, the second condition defines the extend that the fold has on the curve.

For the extraction of subchains that satisfy the first condition we use a simplified version of an orientation based shape matching algorithm introduced in [78].

Let $\mathcal{P}$ be a piecewise linear curve given by $n+1$ vertices $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_n$ according to the definitions is Section 4.2.3 and let $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{n-1}$ be its n segments. Recall that, $\mathcal{L}_k(\mathcal{P})$ the partial length of the polyline's first k segments. To shorten the notation, we refer to it as $\mathcal{L}_k$. The *parametric representation* $\mathcal{P}(t)$ of $\mathcal{P}$ is then given by

$$\mathcal{P}(t) = \frac{(t - \mathcal{L}_i)}{|\mathbf{s}_i|}(\mathbf{v}_{i+1} - \mathbf{v}_i) + \mathbf{v}_i, \tag{5.1}$$

with $t \in [\mathcal{L}_i, \mathcal{L}_{i+1})$ and $i = 0, 1, \dots, n-1$.

The *orientation of a directed line segment* $O(\mathbf{s})$ in the plane is a value in the interval $[-\pi, \pi)$. It refers to the angle the segment encloses with the positive x-axis. Let $\mathbf{s}$ be the directed line segment from point $\mathbf{a}$ to point $\mathbf{b}$, with $\mathbf{a}, \mathbf{b} \in \mathbb{R}^2$. Let x and y be the x-coordinate, y-coordinate respectively, of the point $(\mathbf{b} - \mathbf{a})$. The orientation of the line segment is given by

$$O(\mathbf{s}) = \begin{cases} \arctan(\frac{y}{x}), & \text{for } x > 0 \\ \arctan(\frac{y}{x}) + \pi, & \text{for } x < 0 \text{ and } y \geq 0 \\ \arctan(\frac{y}{x}) - \pi, & \text{for } x < 0 \text{ and } y < 0 \\ \frac{\pi}{2}, & \text{for } x = 0 \text{ and } y > 0 \\ -\frac{\pi}{2}, & \text{for } x = 0 \text{ and } y < 0 \\ \text{undefined}, & \text{for } x = 0 \text{ and } y = 0. \end{cases} \tag{5.2}$$

This function is commonly referred to as *atan2* in many programming languages

The *orientation function of a polyline* $O_{\mathcal{P}}$ maps every point on the polyline $\mathcal{P}$ to the orientation of the current segment. It is piecewise constant and defined as

$$O_{\mathcal{P}}(t) = O(\mathbf{s}_i) \tag{5.3}$$

with $i = 0, 1, \dots, n-1$ and $t \in [\mathcal{L}_i, \mathcal{L}_{i+1})$.

Note that the orientation value of two segments that enclose a small angle, might, however, differ very much. This may happen, in case one segment has a high positive orientation value near π and the other one a low negative value near $-\pi$. This might lead to jumps in the orientation function from $-\pi$ to π and vice versa, resulting in severe discontinuities in the function.

The orientation function is piecewise constant and thus, we technically cannot speak of a continuous function. However, we can make the assumption that the polyline that we examine approximates a smooth curve by dense sampling. Consequently, the orientation value of two adjacent segments is very likely to be similar, and in virtually all cases much smaller than π.

We take this property in order to iteratively achieve a pseudo-continuity of orientation values of a polyline $\mathcal{P}$. The idea is that at a positions with jumps that are larger than π, all function values right of that positions are shifted by $\pm\pi$. Algorithmically, this is achieved in the following way:

1. Let $\Delta = O_{\mathcal{P}}(t_j) - O_{\mathcal{P}}(t_j + \epsilon)$ with $|\Delta| > \pi$ be a jump at a position t_j.
2. If $\Delta > \pi$, set $O(t)$ to $O(t) + 2\pi$ for all $t > t_j$.
3. If $\Delta < -\pi$, set $O(t)$ to $O(t) - 2\pi$ for all $t > t_j$.
4. Repeat, until there are no jumps of adjacent orientation values left.

The resulting function does not suffer from severe discontinuities any more.

The orientation function might be noisy. Hence, we calculate the *smoothed orientation* $\tilde{O}_{\mathcal{P}}$ by applying a low pass filter to $O_{\mathcal{P}}$. High frequencies are eliminated and the number of local extrema of the function reduced.

5.3 MINIMAL SUBSETS

A minimal subset indicates whether a fold is located on the curve. Minimal subsets are portions of a curve where the curve bents at least in an angle of τ. They are given by intervals on the curve's parametrization. We use extrema of the smoothed orientation function for determining these intervals.

Steep slopes between adjacent extrema indicate a strong bending in the original curve. Positive slopes indicate that the curve bends its left, while negative slopes indicate that it bends to its right. Since, our curves have either right or left-oriented folds, the folds of a curve all share the same bending direction at their tips.

An interval that supports a slope is a valid candidate for a minimal subset, if the difference between the extreme values is larger than τ. This indicates, that the course of curve performs direction change of at least τ. All intervals of minimal subsets must support either positive or negative slopes exclusively. If the number of valid subset candidates that result from negative slopes is greater than the number resulting from positive slopes, the former ones are the minimal subsets. If not, the intervals supporting the positive slopes are defining the minimal subsets.

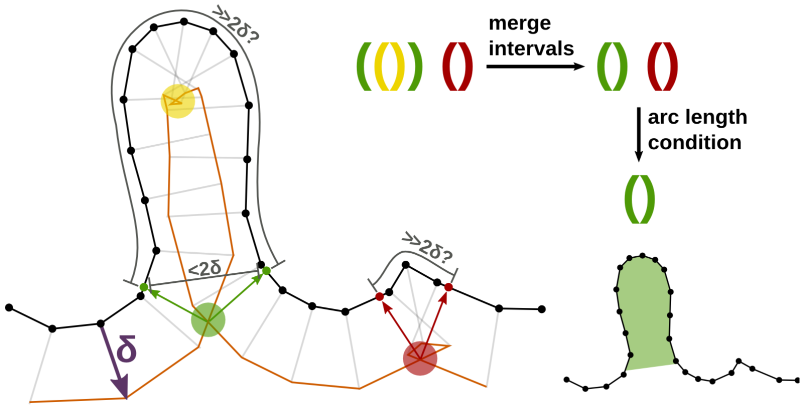


Figure 5.1: Procedure of obtaining maximal subsets. The orange polyline results from offsetting the original membrane line by the value δ. Its self intersections (marked in yellow, green, red) are translated into intervals and overlapping intervals are merged. Intervals whose arc length along the membrane are shorter than 2δ are discarded.

Once the intervals are found on the orientation function, the same intervals are mapped back to the original curve yield the minimal subsets, i.e. parts of the curve where the bending is greater than τ.

We found $\tau = 2/3\pi$ to be a good default parameter that gave viable results in most of our data sets. The determination of minimal subsets also works on polylines that have a strong overall curvature, e.g. curves (with folds) that follow a nearly circular course.

5.4 Maximal Subsets

While the minimal subsets serve to display the positions at which folds are located on the curves, the maximal subsets, by contrast, determine the range on a curve that is covered by a fold. Condition 2 can be satisfied with the aid of self-intersections of *polyline offsets*. Their detection is illustrated in Figure 5.1: first, we apply an offset δ to each point in the membrane polyline in the direction of the weighted average of normals from adjacent segments. This offset line (orange) features self-intersections that denote points on the polyline whose distance is smaller than 2δ. The intersection points are back-projected onto the polygonal chain, which indicate the limits of membrane intervals that fulfill condition 2. There are usually multiple intersections of the offset line within one fold. Hence, we unite overlapping intervals and keep the one that results in the longest subchain of the membrane with end points being less than 2δ apart. A good default value for δ is the width of an average fold.

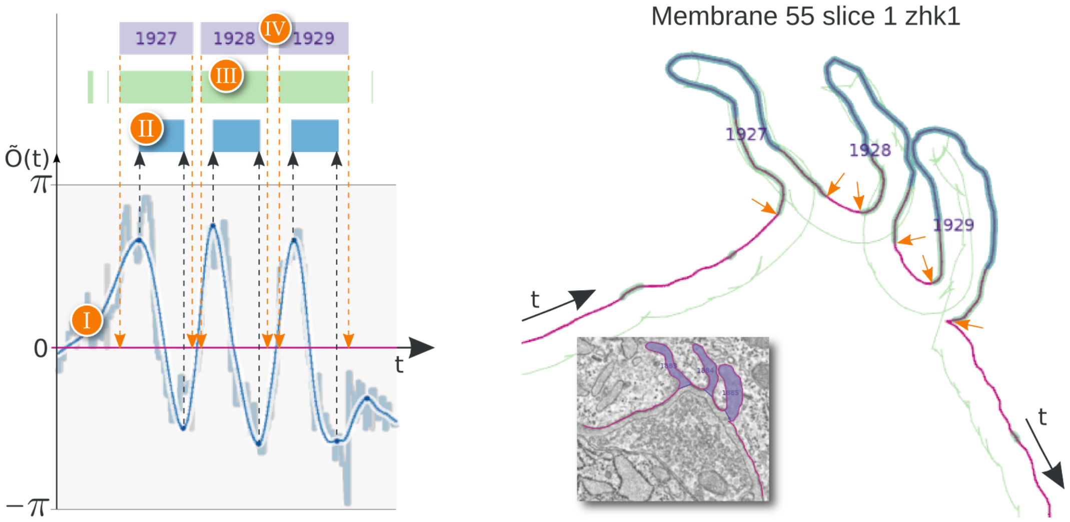
We discard all those intervals whose arc length along the membrane are shorter than 2δ.

5.5 Interactive Parameter Tuning

Three parameters have to be set for the feature extraction: two for the determining minimal subsets and one for the maximal subsets. We propose default values for our parameters that usually work very well on most data sets. Assuming that a data set originates from the same tissue sample and passed through the same segmentation process, it is in general sufficient if the parameters are tuned once for a small part of the data set, e.g., for a single slice of a data volume, and then applied to the rest of the data. Figure 5.2 displays the information that is being shown to users upon tuning these three parameters. We display the orientation function and the intervals of the subsets to make the detection algorithm and the effect of single parameters transparent to the user. This permits an intuitive adjustment of parameters. Users may thus study the consequences of changing a parameter value both with respect to the results as well as with respect to the algorithm. This permits the rapid detection of parameters that need to be changed in order to achieve a certain goal for the fold detection. Moreover, it also facilitates figuring out whether a given parameter needs to be increased or decreased.

5.5.1 Smoothing Factor and Turning Angle

The minimal subsets depend on the turning angle difference τ and on the smoothing factor of the orientation function. The latter again depends on the initial roughness of this function. While changing the smoothing factor, the user can observe the original noisy function overlaid with the smoothed function (see Figure 5.2a, I), with the local extrema of the smoothed function being highlighted. Thus, she can easily choose a suitable value. Manipulation of the turning angle parameter τ is usually not necessary. However, a decrease may potentially result in the disappearance of complete minimal subsets where the vertical distance of a pair of adjacent extrema falls below the threshold τ in the graph of the smoothed orientation function. The minimal subsets are displayed as intervals represented by blue boxes (Figure 5.2a,



(A)

(B)

FIGURE 5.2: The visualization of the influence of the algorithm parameters τ, δ, and the smoothing factor on the extraction result. (A) Orientation function (I), with intervals of minimal subsets (II), intervals of maximal subsets (III), and intervals of final folds (IV). (B) Curve with highlighted subsets according to the intervals (II) and (III) in (A). Arrows indicate corresponding locations.

II). These are plotted above the orientation function, sharing the same x-axis. This clarifies their correspondence to the orientation function and its extreme values, which is also emphasized by the vertical black arrows in the figure. Additionally, the intervals are highlighted in blue on the original polygonal chain, shown in pink in (Figure 5.2b).

5.5.2 POLYLINE OFFSET

The maximal subsets are depicted as intervals represented by green boxes above the orientation function, sharing the same x-axis (Figure 5.2a, III). The influence of tweaking the offset δ can directly be observed in the resulting offset polyline, which is shown in green running in parallel to the pink one in Figure 5.2b. The corresponding areas are highlighted in green on the original polyline, while boundaries are emphasized with orange arrows. A higher offset value results in wider intervals, i.e. folds "grow" more along their sides. Since there is no exact definition of where a fold starts, this choice should be left to the domain expert. With respect to the further analysis of the folds, it is necessary that the offset δ stays fixed for the

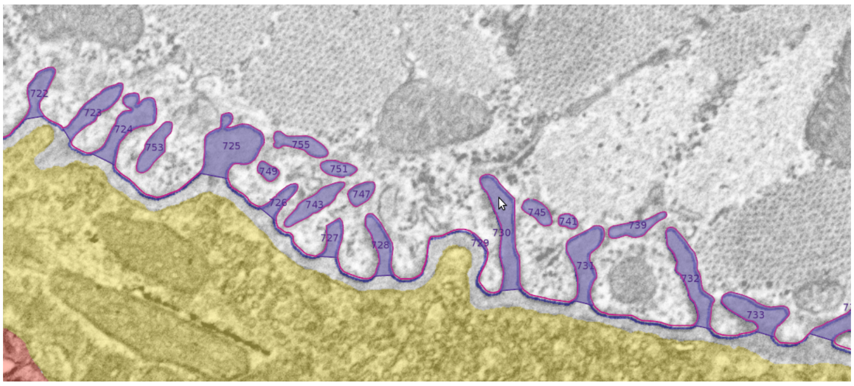


Figure 5.3: Detected folds (violet) on a muscle cell membrane (magenta). A large variety of shapes is covered by the extraction.

whole data. A good default value for δ is the width of folds in the data set, which does not exhibit variations within one tissue sample.

5.6 Junctional Folds

Junctional folds may now be identified by a combination of minimal and maximal subsets. As introduced earlier/explained earlier, minimal subsets tell us whether a fold is present, while maximal subsets determine its extent. Thus, a junctional fold is a maximal subset that contains a minimal subset.

The final folds are labeled with numbers and their interval representation is depicted in violet on top of the orientation function (Figure 5.2a, IV).

The intervals are also highlighted on the polyline by filling the interior of the corresponding folds (Figure 5.2b, bottom).

Note that in general, using exclusively maximal subsets is insufficient to determine folds. This approach tends to overestimate the existence of a fold, which can be observed in Figure 5.2a: in total, six maximal subsets (green blocks, III) are found in this example, whereas minimal subsets (blue blocks, II) have been detected at three positions. Only the combination of both criteria (violet blocks, IV) gives the desired results.

5.7 RESULTS

For most cases the default values work very well for the detection of folds in a polygonal chain and even detected folds of a highly-unusual shapes. Figure 5.3 shows an example of the proposed fold extraction technique on a part of a neuromuscular junction. All folds were automatically detected. The bulge of the neuron (yellow) in the center was correctly considered not to be a fold by our algorithm. If required, the parameters can be adjusted interactively and the result can be judged instantly. The resulting folds are projected onto the microscopy images, permitting the user to judge the correctness of their extraction in their biological context.

We applied the algorithm to the vectorized data sets of the six neuromuscular junctions, that were introduced in Section 4.1. The data sets comprise 78 slices, on which we were able to determine 1392 folds that are located on the membranes. For the determination of secondary folds, we applied the algorithm to the polylines of the resulting folds. On 1141 of these folds, no further folds were found. On 251 folds, however, in total 557 additional folds were detected. According to the fold data classification described in Section 4.3.4, this results in 306 secondary and 251 primary folds. In total we were able to extract 1698 folds from our data sets by means of the introduced algorithm.

5.8 CONCLUSION

In this chapter, we presented an algorithm that detects fold-like features on polylines from segmentations of muscle cell membranes of neuromuscular junctions. We combined two criteria that define a fold and used these for the extraction of the folds. We provide a strategy for interactive, intuitive and informed parameter tuning by visualizing the impacts that the parameters have on parts of the detection algorithm. We proved the viability of the algorithm by applying it to real world data from segmentations of neuromuscular junctions.

6 Proofreading and Correction of Segmentations

This chapter is based on two peer-reviewed publications [9, 8] of the author. The former was presented at the *Vis in Practice* event at IEEE Vis 2015 in Chicago and concentrates on the applied quality assessment method. The focus of the latter lies on the abstract representation of the data and its manipulation.

Challenging biomedical image data is still often segmented manually. To reduce costs and optimize time required, this task is often assigned to non-experts for large-scale data. Their segmentations have to be proofread and corrected in a post-processing step by experts. Especially for large-scale image stacks with intricate segmented structures, this is a tedious task that can greatly benefit from from a tight integration with visual analysis methods. To date, however, only few tools exist to support this task.

In this chapter, we demonstrate how visualization can support the proofreading and correction of segmented surfaces in 3D image stacks (Figure 6.1, left). We present NeuroMap, an interactive framework for proofreading and correcting segmented image stacks of muscle tissue volumes including neuromuscular junctions (NMJs).

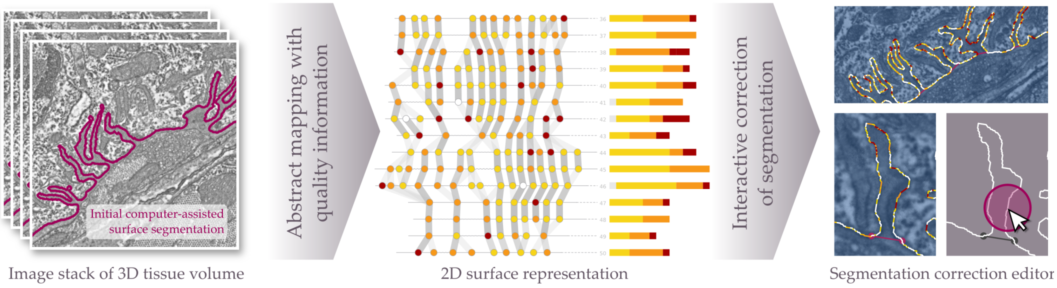


Figure 6.1: NeuroMap supports domain experts in proofreading and correcting stacks of segmented microscopy image data (left). The surface map (center) gives the user a concise 2D overview of relevant structures in 3D including error quantification. Based on this information, the user can navigate to erroneous parts and directly edit them (right).

The centerpiece of this framework is the *surface map* (Figure 6.1, middle), an abstract 2D representation of the volume, which takes advantage of the characteristic ultrastructure of NMJs and aids in the visual communication of data quality. Automatic quality assessment of the segmentation is used to visually guide experts to erroneous parts of their data. In an interactive setup that includes the raw image data, detected errors can be corrected rapidly (Figure 6.1, right). This proofreading pipeline was developed in close collaboration with domain experts in data mining and biology. We demonstrate the utility of the novel framework using segmented image stacks of micrographs of neuromuscular junctions in mice.

6.1 Introduction

Ultrastructural examination of tissues, cells, and microorganisms plays a vital role in current biological research. Such data can be acquired using electron microscopy, resulting in image stacks consisting of hundreds of images per stack that are very rich in detail and textures. The subsequent morphological analysis of such structures relies critically on a reliable *segmentation*.

Obtaining a qualitatively satisfying segmentation is very difficult due to the complexity of the images, noise, and imaging artifacts. Hence, fully automatic segmentations of such image stacks remain difficult.

A common approach to rapidly obtain relevant structures in such data are computer-assisted segmentations by non-experts. Thus, initial results can be obtained at low costs with respect to time and money, as many people can work in parallel and it requires little training, i.e., no expensive specialists need to be involved. The downside of this approach, however, is that the data has to be validated in a second step. This pipeline is only feasible if the costs for the validation are substantially lower compared to the segmentation costs.

While image segmentation is a well-accepted problem with lots of active research and available software (both open-source and commercial), the support for the validation of segmentation results is still limited. Notable advances have been made in recent years [31, 42, 45, 33] (see Section 3.1), but most of them address region-based image segmentations and cannot easily be extended to the extraction of complex shaped surfaces from image stacks.

	segmentation method	segment shape	object type of 3D reconstruction	number of objects	Analysis subject
[31], [42], [45], [33]	pixel-based, region-based	region(s)	solid volume	multiple	Functional interaction between objects
Our project	boundary-based	curve(s)	surface	one	Inherent structure of object (= surface)

Table 6.1: Comparison between our project and other segmentation proofreading/correction/abstraction approaches: the disparities in data setup and types, as well as the different analysis target demand and permit alternative treatments.

Section 3.1 discusses existing approaches for the proofreading process of segmentations. Table 6.1 gives an organized overview of the differences between existing proofreading approaches and ours. They mainly result from different types of the segmented subjects they rely on.

6.2 The Components of NeuroMap

We start with introducing each of the different components of the NeuroMap framework. Subsequently, we provide a comprehensive evaluation of the components and their functional interaction.

Figure 6.2 shows the individual steps and interim results of the segmentation and surface reconstruction task. The gray pipeline on top represents the main flow from the input image stack, by means of a slice-wise segmentation process, towards a high quality segmentation, that is e.g. suitable for surface reconstruction. The quality enhancement loop is shown in purple. It is the iterative process of automatic quality assessment and computer-aided segmentation correction. In blue, we see additional image data injected into the main loop for the unsupervised quality assessment.

The initial segmentation (I) usually suffers from insufficient quality because of the challenging image data. The sarcolemma, the muscle cell membrane we are particularly interested in, cannot be segmented completely automatically as (i) the signal-to-noise ratio in the images is low and (ii) interruptions and density fluctuations of the membrane image, resulting from the specific 2D sampling of the 3D topology, are frequent. Thus the segmentation is done manually or semi-automatically.

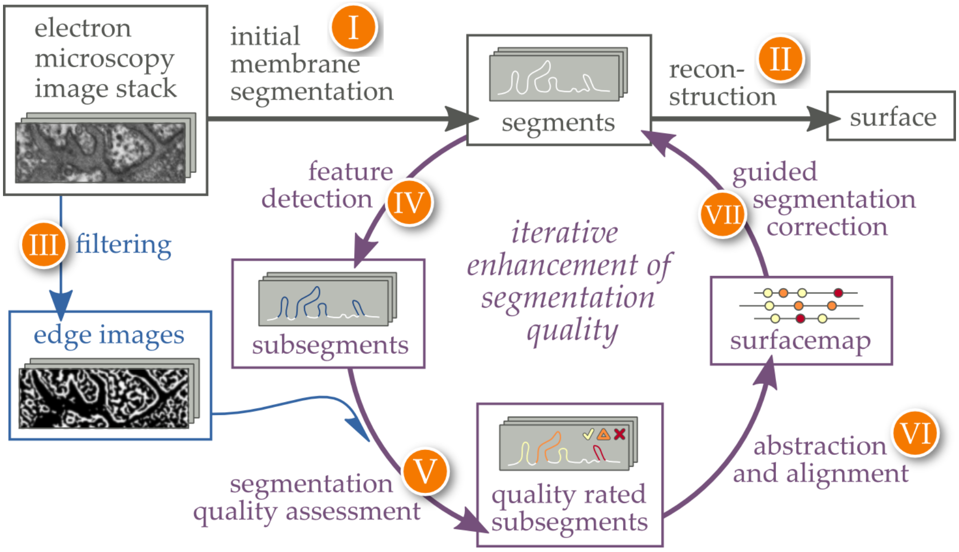


Figure 6.2: The collaboration between the different NeuroMap components. NeuroMap supports iterative enhancement of surface segmentations.

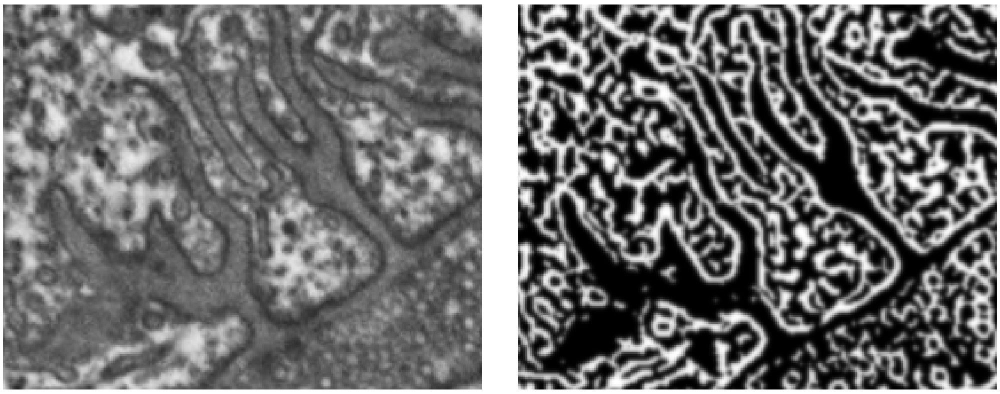
For the reconstruction of surfaces from planar curves (II) applicable approaches, for example by Wang and Tang [47], exist. Hence, it is not covered in this work any further.

We begin the introduction of the NeuroMap framework with the quality quantification (III) and (V), both addressed in Section 6.3. For the decomposition of the segmented sarcolemma curve into fold and non-fold parts (IV) we use the method that we already presented in Chapter 5.

The *surface map* is introduced in Section 6.5, the centerpiece of the proposed framework. The last component to be detailed is the *segmentation correction*, which is covered in Section 6.6.4

6.3 Segmentation Quality Assessment

Quality control of image segmentation results is commonly considered a challenging topic and no standard procedure exists [79]. Existing methods are divided into three classes depending on the type of evaluation they perform: (i) analytical methods that directly examine the segmentation algorithm itself, (ii) empirical goodness (unsupervised) methods that quantify automatically how well certain so-called goodness properties are achieved, and (iii) empirical discrepancy (supervised) methods that compare the segmentation to ground truth data [80]. The most



(A) (B)

Figure 6.3: (A) Detail of original image. (B) Same detail after applying a filter to emphasize edge-like structures introduced by [38].

widely used methods fall into the supervised category where the segmentations are either judged by an expert or are automatically compared to expert-generated ground truth [39]. Such methods, however, are infeasible for large-scale data as they take too long and are too tedious. Here, we present an unsupervised evaluation approach that is embedded into our generic proofreading pipeline.

We now briefly introduce some notation for the subsequent sections. The quality $Q : \mathscr{C} \to [0,1]$ is a function that maps each point on a curve $\mathscr{C}$ to a quality value. Low values indicate poor quality, while high values correspond to excellent quality. This *total quality* is a combination of different *quality measures*. In the following we propose two possible quality measures: The *objectness* $\overline{o}$ and the *simplicity* χ. The total quality Q can be extended by additional quality measures and refined easily, without affecting the functionality of the proofreading framework.

6.3.1 Objectness Filter

For a curve and the record of the sarcolemma, we compute the likelihood $\overline{o}$ that a position of the curve belongs to the sarcolemma using unsupervised segmentation quality assessment.

Looking at a sample from our data (Figure 6.3a) we see that the sarcolemma can be usually quite clearly identified by humans and forms a distinct dark gray edge in the image.

By means of the objectness filter described by Antiga et al. [38] on an inverted, median-filtered raw image with the object dimension set to 1, we can recover the edge faithfully (Figure 6.3b). Antiga's objectness filter exploits geometric properties of the eigenvalues of the Hessian matrix at each pixel location in order to emphasize line-like structures in each of the 2D slices of the 3D image stack. We used a regularization scale of $\sigma = 1$ and the user-defined weighting parameters were set to $\alpha = 1.0, \beta = 1.0, \gamma = 0.1$. The response map of the objectness filter is finally binarized using a conservative threshold setting that does not miss any of the desired boundaries. To account for uncertainty towards the boundaries of the edge, we employ Gaussian smoothing to the filtered image ($\sigma = 1$). As we can see in Figure 6.3b, the filter is not specific to the sarcolemma, but detects a large amount of similar structures in the microscopic images. Hence, the objectness image can tell whether a pixel belongs to an edge, but cannot be directly used for automatic segmentation as discrimination of the different edge types is not (yet) possible.

6.3.2 Objectness Measure

The objectness image can now be used to identify parts of the segmentation that do not belong to edge structures in the image. To this end, we compute the objectness $\overline{o}$ under the segmented curve.

For piecewise linear curves this can be done line segment-wise: The objectness is computed for each line segment of the polyline that represents the sarcolemma. Let $O = (O)_{i,j}$ be the objectness-filtered image. Let $T = (T)_{i,j}$ be a texture with the same dimensions as O and with entries $t_{ij} = 0$. Moreover, we assume that the line segment **s** is rendered anti-aliased (width: 1 pixel) into the gray-scaled texture T_s. Its entries t_{ij} now represent the fraction of presence of the segment. The objectness $\overline{o}$ of the line segment is given as

$$\overline{o}(s) = \frac{\sum_{i,j}\left((O)_{i,j} \circ (T)_{i,j}\right)}{\sum_{i,j}(T)_{i,j}}, \tag{6.1}$$

where $\circ$ denotes the Hadamard product, which is calculated entry-wise.

In other words, the line segment of the polyline is rated by the mean of the intensity of its underlying pixels (of the objectness image). Thus, polyline segments that are oriented along white edges are considered to be of higher quality than those that cross dark, i.e., non-edge regions.

6.3.3 Simplicity Value

Another criterion that influences the segmentation quality is the *curve simplicity*. A planar curve $\mathscr{C} : I \to \mathbb{R}$ ($I \subset \mathbb{R}$) is called *simple* if it has no self-intersections, i.e. the mapping $\mathscr{C}$ is injective (which also precludes self-superposition).

The sarcolemma is a two-dimensional surface in the three-dimensional space. The segmentation curves follow cross sections of this cell membrane. Thus, cases of self-intersection and overlays are impossible when following the course of the sectioned sarcolemma. In real-world data, such situations commonly result from segmentation errors. In particular, overlaps occur occasionally in segmentations created using the interactive Livewire technique. Thus, the simplicity value $\chi \in \{0, 1\}$ penalizes regions of superposition: unique points on a curve $\mathscr{C}$ are mapped to one, while non-unique points are mapped to zero.

For piecewise linear curves, segments that intersect or overlap either completely or partially are assigned a simplicity value of zero in their entirety.

6.3.4 Total Quality

Having discussed the individual components of the quality function, we are now capable of giving a definition. Unless specified otherwise, we will be using

$$\begin{aligned} Q : \quad \mathscr{C} &\to [0,1], \\ \mathbf{s} &\mapsto \bar{o}\chi \end{aligned} \tag{6.2}$$

as function of the total quality.

A combined quality value for multiple line segments segments (e.g. folds or entire slices) of piecewise linear curve segmentations results from summing up individual quality values weighted by the length of their respective line segments.

6.4 Requirements Analysis

NeuroMap is designed for proofreading and correcting segmented surface data. The targeted users are experienced domain-experts (biologists) who are familiar with interpreting microscopic images. Based on informal inquiries among biology experts we obtained an impression about their current practices and demands.

Their conventional approach is to examine the segmentation in the image stack slice by slice. At ambiguous spots (where the image quality might be poor) adjacent slices are consulted, which requires going back and forth in the image stack. Analyzing the experts' current routine, we distilled the following requirements for the framework:

R1 View the source data. A necessity that is often demanded by domain experts is the link between abstract representations that are based on their data and the view on the source data itself. This confirms the confidence in the representation because the abstraction can be verified by means of the source data (at least on a sample basis). Furthermore, it enables them to explore details that are not shown in the abstraction.

R2 Find erroneous parts of segmentations directly. Instead of successively inspecting all segmented structures from start to end, a proofreader's attention should be directly drawn to areas with qualitatively poor segmentations. This leads to an increase of segmentation quality, which simultaneously permits increasing the homogeneity of the quality.

R3 Edit the segmentation. Here, correcting the segmentation means editing of curves represented by polylines. This requires the presence of the original image data (compare R1) in order to provide a guidance for the editing person.

R4 Do not get lost in details. For inspecting single segmentation portions in the source data, one might heavily zoom into the original data, viewing a particular slice of the volume. In this situation, it is important to get answers to questions such as "Where am I are in my volume?", "How much quality issues remain to be inspected?", or "Does this fold have a corresponding part in adjacent slices, or is the segmentation maybe based on an artifact?".

6.5 SURFACE MAP

The most challenging requirements are R2 and R4 as they require a novel interface to the data. To address them, we designed the *surface map* that provides an abstract overview over the complex 3D surface data. Furthermore, it can also be used as a means to easily access the image data. For this approach we structure the surface by its own characteristic features. This guarantees an intuitive mapping between

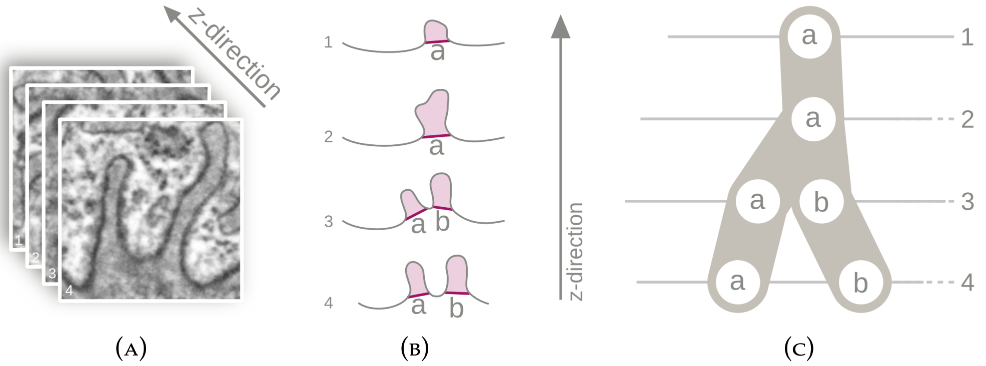


Figure 6.4: The concept of the surface map. (A) This illustrative example shows a stack of four images of adjacent slices. Extending through the stack, the sarcolemma evolves from one single fold to two folds. (B) This is shown in the four segmented cross-sections of the sarcolemma, with detected folds shown in purple. (C) These fold structures are summarized in the surface map.

the abstraction and the image-based segmentation (and even the plain microscopic images).

A characteristic feature of a cross-sectioned sarcolemma at a neuromuscular junction is the shaping of folds. A junctional fold is an invagination of the muscle cell membrane with a width of about 0.1 to 0.2 µm and a depth of 0.5 to 1.0 µm. Often, these folds are not straight but bent. This bending is the main source of the membrane's structural complexity. The main idea is to simplify the surface by truncating the folds. For this purpose, we subdivide each segment (which is represented by a curve), into smaller parts (subsegments), each either forming a fold or the link between two neighboring folds.

For this purpose, we subdivide each segment (which is represented by a curve) The method we use for the automatic identification of folds is described in [10].

6.5.1 Design

Figure 6.4 depicts the surface map of a small subsection of the data. Each image in the input image stack corresponds to a horizontal line in the surface map. In each such image, the intersection between imaging plane and sarcolemma forms a curve (Figure 6.4b) with extensive folding. In the surface map (Figure 6.4c), each fold f is represented by a node n_f. Membrane segments between folds are represented by thin horizontal lines between the nodes.

The length of the gap between two adjacent nodes is proportional to the Euclidean distance between their corresponding folds. The nodes are connected by horizontal edges. An edge is not drawn straight, but given a wave-like appearance. For this purpose we use quadratic Bézier curves to approximate a sine wave with a fixed phase. The magnitude of the edge's tortuosity depends on its *fold distance ratio*, which is the arc length of the subsegment located between both folds normalized by their Euclidean distance. The influence of this ratio on the edge can be observed, for example, in Figure 6.10 (page 79) on edges which we labeled with (c).

Folds commonly form ridges in 3D that span many images, i.e., there is a correspondence of nodes across multiple horizontal lines (which correspond to the images) in the surface map. Comparing a segmented fold across multiple slices helps in the quality judgment. Often the images suffer from noise and boundaries are barely pronounced. In these cases, automatically highlighting the connections of folds across multiple slices is advantageous for the correcting expert. We represent those connections by edges between two nodes from adjacent rows. The automatic estimation of likely connections will be detailed in Section 6.5.3.

6.5.2 Encoding of Fold Properties

Each individual fold is represented by a circular node in the surface map. These nodes can be colored according to any property of the folds, which permits the inspection of spatial distribution of different fold features such as the depth of folds. In our particular case, the proofreading and correction of surface segmentation, we will use the node colors to encode the segmentation quality of each individual fold.

6.5.3 Inter-row Alignment

The positioning of the rows, i.e. slices, in the surface map is an optimization problem. While we have a good prior for the horizontal distances of nodes, i.e. folds, within the same image (which equals the Euclidean distance between two folds in the image), finding a good alignment between the rows is more difficult. Two properties contribute to the complexity: (i) The surface of the sarcolemma is not planar and the projection to 2D introduces distortions. (ii) There is no well-defined start and end point for the segmentation in each image. Hence, we need an alignment procedure to position each horizontal line such that it carefully reflects the topography and still gives a good idea about the local connectivity. In this section, we will solve the problem of finding corresponding folds across images. We will use this information to position nodes and their links in the following.

CORRESPONDENCES

Connections exist between folds of two adjacent slices, $slice_i$ and $slice_{i+1}$. Let F and G be the (ordered) sets of folds of $slice_i$, and $slice_{i+1}$ respectively. Let $C := F \times G$ be the set of all (theoretically) possible connections with elements $c := \{f, g\}$. Furthermore, let $p(f) \in \mathbb{R}^2$ be the spatial position of fold f in the image source. Each connection $c \in C$ has a certainty value $w \in [0, 1]$ that describes the likelihood of the two involved folds' association. We consider two geometric properties for approximating those likelihoods: (i) the Euclidean distance between the centers of the folds on the baseline, and (ii) the ratio of the folds' intersection areas. Better results were obtained using method (i) as the slice thickness is often still too large to capture follow-up shapes that match well.

The likelihood based on the Euclidean distance is computed as

$$w(c) := \max\left(1 - \frac{\|p(f) - p(g)\|_2}{d_{max}}, 0\right), \tag{6.3}$$

where d_{max} is the maximal accepted distance between two folds that may still be considered to be a connection.

We reduced the amount of *all (theoretically) possible* connections C, whose elements intersect each other, which is not possible in reality, to a set of logically reasonable connections $\tilde{C}$:

$$\tilde{C} := \bigcup_{C_h} \max_{c \in C_h} w(c), \tag{6.4}$$

with $C_h := \{c \in C \mid h \in c, \text{ with } h \in F \cup G\}$. This ensures that for each fold only its connection with the highest certainty value, plus one potential additional connection remains. This additional connection might be the one with the highest certainty value of *another* fold. This leaves us with a maximal number of $|F| + |G|$ connections.

To communicate the probabilities of connections, we use transparency for the inter-slice connections. Making their transparency $\alpha \in [0.2, 1]$ dependent on the certainty by $\alpha(w) := 0.8w + 0.2$, ensures the visual emphasis of connections with high certainty values, without losing alternative possibilities with lower certainty values.

ALIGNMENT

Now that we also know the inter-slice connectivity of nodes in the graph representing our surface, we can address the problem of spatial positioning. The horizontal alignment of the rows is determined by minimizing the slants of the connection representatives. The algorithm operates iteratively: row_{i+1} is positioned with respect to its predecessor row_i by shifting it horizontally by an offset δ. REcall that, $p(f) \in \mathbb{R}^2$ was defined to be the spatial position of fold f in the image source. The parametric location of a node n_f along the abstract membrane representation is denoted as $t(n_f) \in \mathbb{R}$. The alignment of a row is determined by minimizing the summarized signed skewness $s(c) := t(n_f) - t(n_g)$ of each connection $c = (f, g) \in F \times G, f \in F, g \in G$ between the current row and its predecessor. For the summation we assign each connection a weight that corresponds to its certainty value $w(c)$.

$$\delta = \frac{\sum\limits_{c \in C} s(c)w(c)}{\sum\limits_{c \in C} w(c)}. \tag{6.5}$$

As detailed before, we use an iterative algorithm going from top to bottom and adjusting the position of the current row with respect to the previous one. The result of this alignment is shown in Figure 6.11c on page 81.

6.5.4 MEMBRANE PROPERTIES REQUIREMENTS

The construction of the surface map is invariant with respect to rotations or to the membrane's main curvature. This implies that, there is neither a calculation of a main axis involved, nor a limitation of the membrane's main-curvature is set. The only requirement concerning the shape of the sarcolemma's cross-section is that it has to feature several folds, as is the case near a neuron. Otherwise, there would be no features at which the sarcolemma could be subdivided and the surface map would simply result in a stack of horizontal lines.

6.6 THE NEUROMAP FRAMEWORK

Figure 6.5 gives an overview of our analysis framework consisting of multiple views that provide complementary information about the data. In the following we will first look at the view depicting the input data (slice view). We then continue with

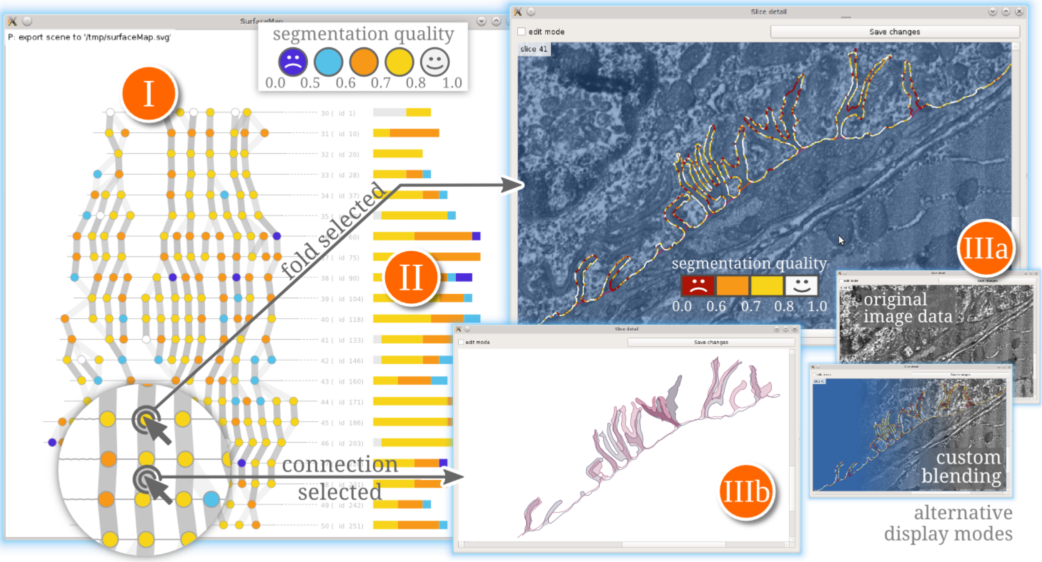


Figure 6.5: GUI overview. I: Surface map. II: Summary chart. III: Slice view, (a) slice mode, (b) connection mode.

the abstract surface representation (surface map), and finally we conclude with the quality summary chart. A video available under https://vimeo.com/portl/neuromap (password: `nmjmap`) illustrates how to interact with NeuroMap and how its different parts are connected with each other.

6.6.1 Slice View

The *slice view* provides users with direct access to their data. It depicts the image stack and the slice-based segmentation, providing standard functionality such as zooming, panning, and interactive hiding of elements.

We distinguish between two modes: (i) The *slice mode* displays information focused on a single slice of the image stack. We currently use microscopy data combined with segmentation results (Figure 6.5, IIIa). The segmentation curve is color-coded according to the local quality criterion as discussed in Section 6.3. The blue tint that is applied to the originally grayscale microscopic image makes for a better visibility of the segmentation, particularly when it is drawn multicolored.

(ii) The *connection mode* (Figure 6.5, IIIb) helps the user to analyze the connectivity of the surface across two consecutive slices. For this view we confine the display to the subsequent segmentation polylines and their fold markers. The strength of this view lies in the easy validation of the suspicion that a fold has been missed in the segmentation or has been falsely segmented.

6.6.2 Surface Map

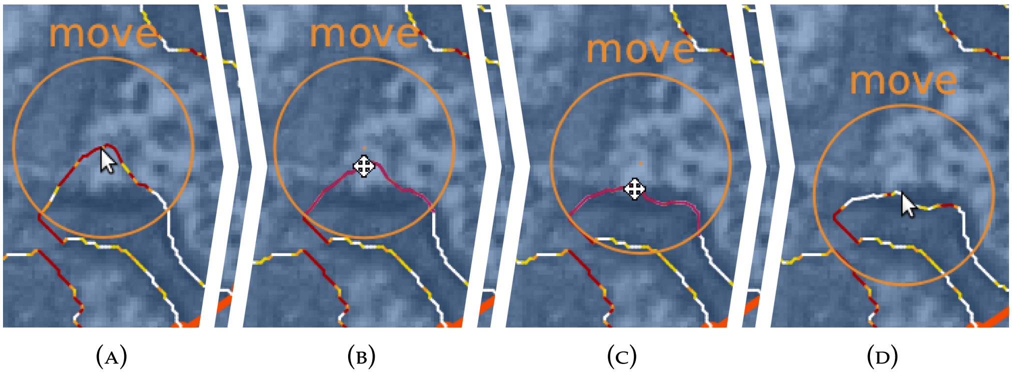
The surface map as presented in Section 6.5 is used to provide an overview over the entire data and to help the user to quickly navigate in the image stack. Additionally, it can serve for annotation purposes, which we currently handle externally, using for example Microsoft PowerPoint or simply a printed version. This is helpful for taking notes about large-scale properties of the data such as data problems or interesting areas.

However, the surface map does not only give a concise summary of the sarcolemma's topography, it also is a crucial interface for the error correction. The expert can easily get information about the entire surface and we further enrich this representation by an estimated error code. For each fold we compute the likelihood of a correct outline using the quality measure introduced in Section 6.3. The resulting quality values are mapped to the fold representations in the surface map using a discretized color map.

Both views, slice view and surface map, are linked with each other. The user can select any element in the surface map and is led to the respective parts in the image stack. For example, a user selects a node in the surface map and is shown the highlighted fold in the respective image. The role the surface map plays within the framework and its interactivity is illustrated in Figure 6.5 and in the supplementary video.

6.6.3 Quality Summary Chart

The *summary chart* (see Figure 6.5, II) condenses the segmentation quality per slice. It is the coarsest display of the data in our framework. The number of folds per slice and the determined quality of their segmentation is encoded by means of a stacked bar chart. The length of a bar corresponds to the number of folds in the respective slice. The ratios between the different quality levels of folds in a slice are represented in its colors.



(A) (B) (C) (D)

FIGURE 6.6: Correction of the segmentation with the segmentation editor. (A) Hovering over the polyline creates a circle around the nearest vertex. (B) and (C) Manipulated polyline while dragging. (D) Updated polyline.

6.6.4 CORRECTION EDITOR

While the last three components of NeuroMap (Slice View, SurfaceMap, Quality Summary Chart) are used for data proofreading, we also found it necessary to be directly able to correct erroneous data in the same software. Hence, we integrated segmentation correction facilities for the intuitive and rapid correction of polylines into the slice view (on slice mode).

Simple drag operations performed on single vertices of the polyline are insufficient for segmentation correction, since it is unwieldy to drag whole ranges of the polyline while trying to preserve smoothness [81]. To overcome this problem, we deform consecutive parts of the polyline within an interactively defined radius, the *affect radius*. Hovering with the mouse pointer over the polyline triggers the appearance of a circle around the closest polyline vertex. The radius of this circle can be adjusted by scrolling. All vertices directly before and after v_i that are in the interior of this circle lie within the affect radius. They are translated into the motion direction of the pointer. This process is illustrated in Figure 6.6 and in the supplementary video. The translation magnitude of the selected vertices depends on their distance to the center of the circle v_c. Longer distances result in larger attenuation values. The factor of a vertex's v_i translation magnitude is determined by the function $\text{att}(\cdot)$. This attenuation function has to satisfy the following requirements: (i) featuring compact support, i.e., it has to map to zero, outside a given finite interval, (ii) $\text{att}(\cdot) \in C^1$, i.e., it is continuously differentiable. Requirement (i) is due to the constraint that the vertex translation is restricted to vertices inside of the effect radius. The second requirement ensures the preservation of the polyline's (visual) smoothness.

A common choice to model attenuation would be using a Gaussian kernel. However, this kernel does not provide a compact support, so we constructed a cosine-based attenuation function. It satisfies our requirements and depends directly on the radius r of the effect circle and its center vertex v_c:

$$\operatorname{att}(v) := \begin{cases} \frac{1}{2}\cos\left(\frac{\pi}{r}\|v - v_c\|_2\right) + \frac{1}{2}, & \text{if } \|v - v_c\|_2 \leq r \\ 0, & \text{else} \end{cases} \tag{6.6}$$

This editing approach guarantees a maximum amount of manipulation freedom while preserving the smoothness of the polyline.

An additional editing mode involves erasing parts of the polyline. While in removal mode, vertices that lie within the effect radius are erased.

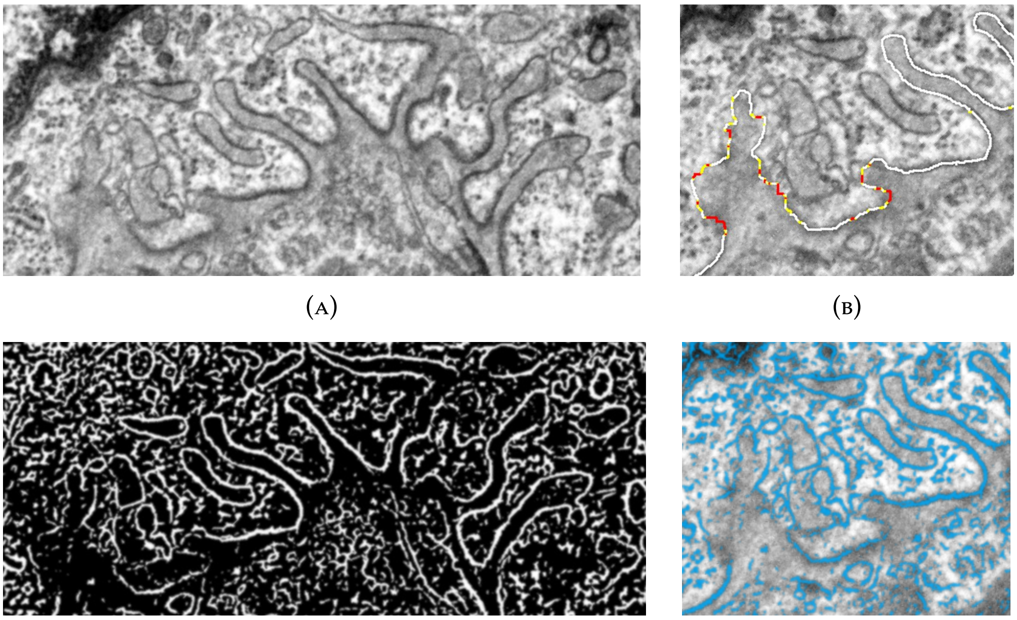
In both editing modes, new vertices are automatically inserted to subdivide polyline segments that exceed a given length after the performed editing action (for vertex translation: press → move → release; for vertex deletion: click). Segments are subdivided to approximately adjust their length according to the polyline's mean length of line segments.

Lastly, we grant multi-level undo and redo functionality during the editing process to ensure a stress-free user experience.

6.7 RESULTS

The NeuroMap software was used for the analysis of six data sets as described in Section 4.1. Our domain experts stated that they never had the possibility to examine it in such a clear and structured view before.

For the evaluation we use different subsets of the data, that are segmented manually using open-source vector graphics software (Inkscape) and semi-automatically based on the *Livewire* method [37] that is implemented in the segmentation error control software supported by the OpenCV library [82]: to enhance the boundaries in the microscopy images, we apply the objectness filter [38], which we also use for error control. Based on the edge-enhanced image, we employ a Livewire technique by Stegmaier et al. [7].



(a) (b)

(c) (d)

Figure 6.7: Segmentation quality analysis for the mdx data. (a) Original microscopy data. (b) Segmentation quality ($Q = \overline{o}$). (c) Objectness image. (d) Overlay of original and objectness image.

The results were similar and we will give a detailed analysis for some subsets of the data. For the entire evaluation process, we concentrate on subsets that feature extensive amounts of folds.

6.7.1 Quality Analysis for a Single Segmentation

We will now investigate the performance of our the surface map. To this end, we will first look at the segmentation quality of an mdx dataset and identify regions of poor data quality and errors in the segmentation. In the second example, we will demonstrate how our method can be used to compare segmentation techniques.

The first dataset comprises 17 images of an mdx mouse where a single contact area between neuron and muscle cell was segmented (Figure 6.7a). The total image stack size is 59. The surface contains six finger-like folds that were segmented on all slices using the Livewire approach. The number of folds varies between slices due to topographic changes. The data was segmented by non-experts. Segmenting each image took less than one minute.

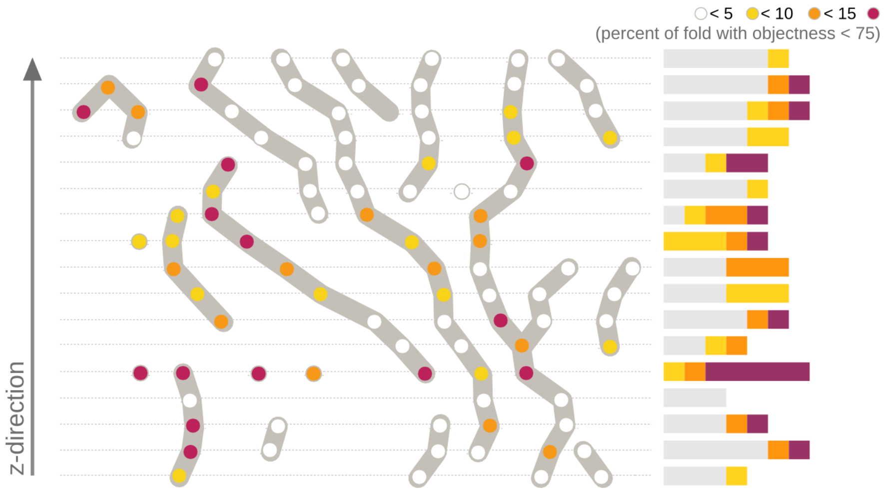


FIGURE 6.8: Conceptual evaluation of the surface map[1] on 17 slices.

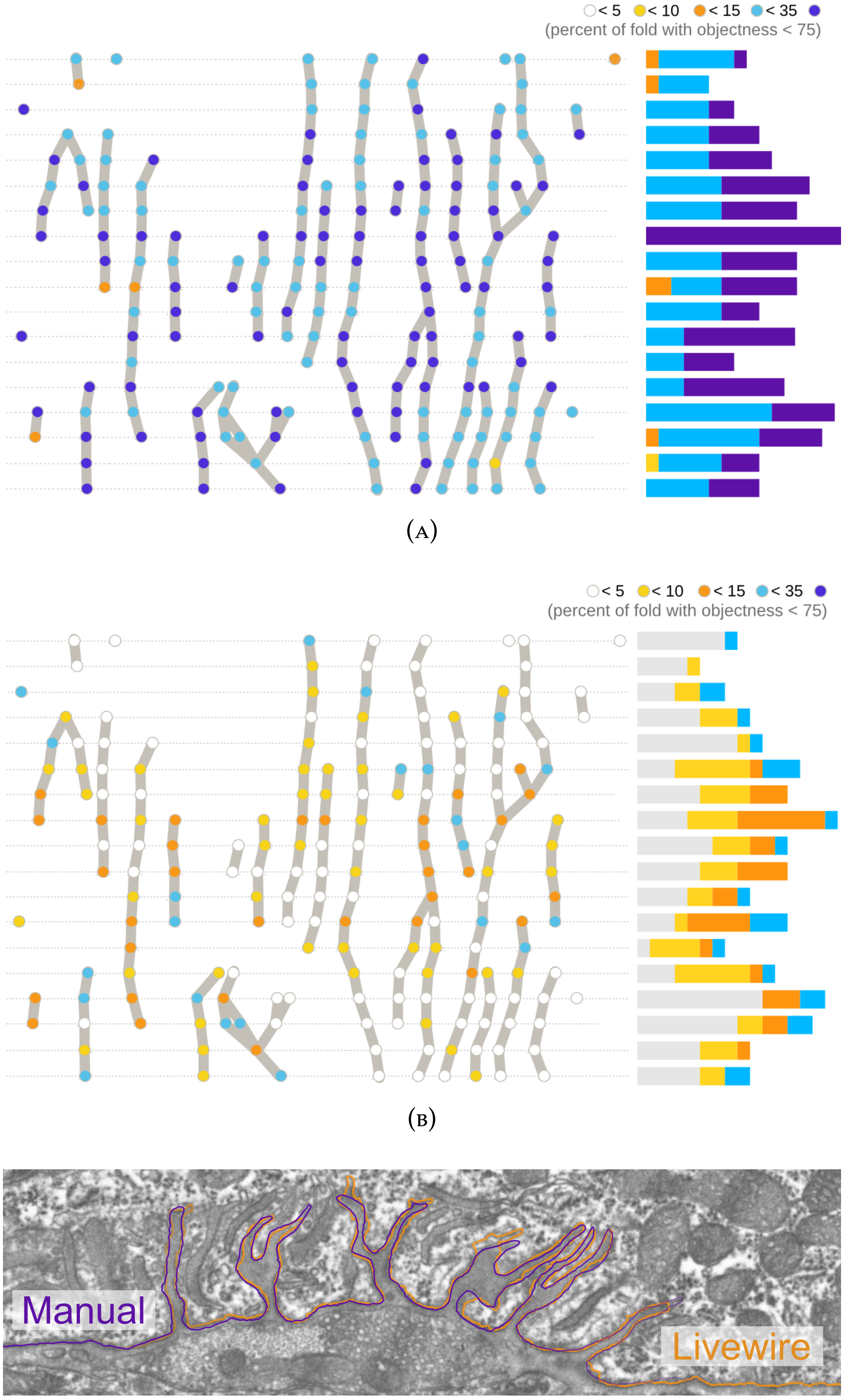
From the sample image, we can see that the synaptic fold is commonly clearly visible. The objectness image (Figure 6.7c) shows that the edge can also be clearly identified by the computer. Poor boundary discrimination can be observed at the wide fold in the lower left corner. This finding is reflected in the overlay image (Figure 6.7d) where the well-defined boundary vanishes. Using the error quantification described in Section 6.3 we can color-code the segmented curve in the image (Figure 6.7b). Using red color, we can rapidly draw the attention of users to regions with poorly-defined boundaries and hence, uncertain segmentation.

Looking at the surface map (Figure 6.8), we see that, for the most part, the segmentation is of high quality. The majority of folds is colored white meaning that less than five percent of its outline falls in regions with very low objectness. Large errors can be observed at the folds on the left-hand side, where orange and red colors dominate the quality. Investigating these folds, we found that the low quality is due to low data quality, i.e., poorly-defined boundaries.

To check the global quality we can also consult the quality chart on the right-hand side of Figure 6.8. We see that the quality distribution is fairly consistent. Slices 6, 7, and 12 stick out due to poor quality. Here we found that the errors are not due to sloppy segmentation of that particular image, but that the quality of these images is much worse in comparison to the other images in the stack.

---

[1] In this example the layout of the surface maps is not algorithmically arranged, but manually. These sections present the results of an early stage evaluation of the surface map concept. However, the distances between of adjacent folds are proportional.



(A)

(B)

(C)

Figure 6.9: Comparison of the surface maps[1] of semi-automatic Livewire (A) and manual (B) segmentation. (C) An overlay of both segmentations (the six right-most folds of slice 5) reveals the differences.

6.7.2 COMPARISON OF SEGMENTATION TECHNIQUES

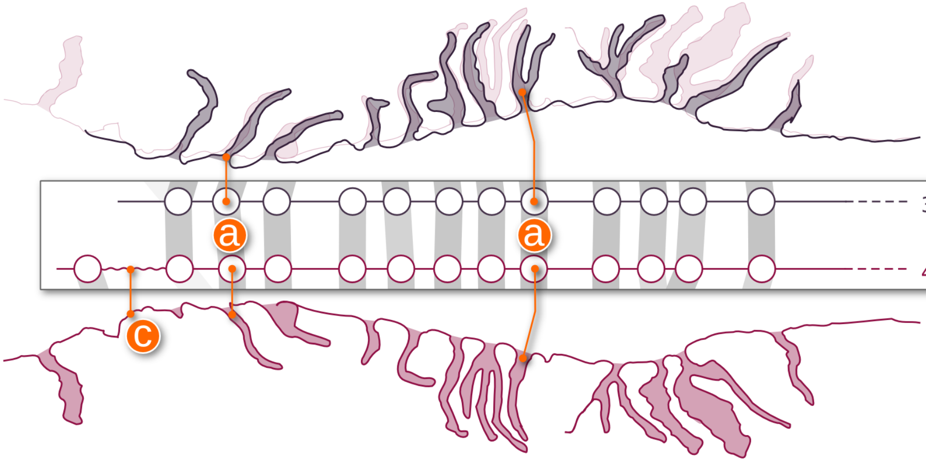
Next we want to compare the quality of competing segmentation techniques. We segmented the same subset of data (18 images of a wild-type mouse) using manual and Livewire segmentation. Again the segmentation was performed by non-experts and validated by an expert. The Livewire segmentation took less than two minutes per image. The time for manual segmentation was considerably longer (approximately four minutes and heavily depends on the desired accuracy). As described in Mortensen et al. [83], we can confirm that manual segmentation is slower and significantly more demanding than the Livewire segmentation. They also showed that the Livewire approach is more reliable when non-experts segment images (small sample set ranging from natural scenes to medical images). Figure 6.9 depicts the results for our data. While error rates are commonly below 15 % for the Livewire segmentation, most errors in the manual segmentation are above this value. This finding is fairly consistent for larger samples. We can also see that the large-scale errors are distributed equally in the image. This demonstrates that the segmentation performance was very consistent during this session, and that segmentation errors occur frequently in the given setting. Figure 6.9c illustrates this findings where we can see that the manual outline frequently leaves the sarcolemma by a few pixels.

6.7.3 VALIDATION OF THE CONNECTION ASSIGNMENT

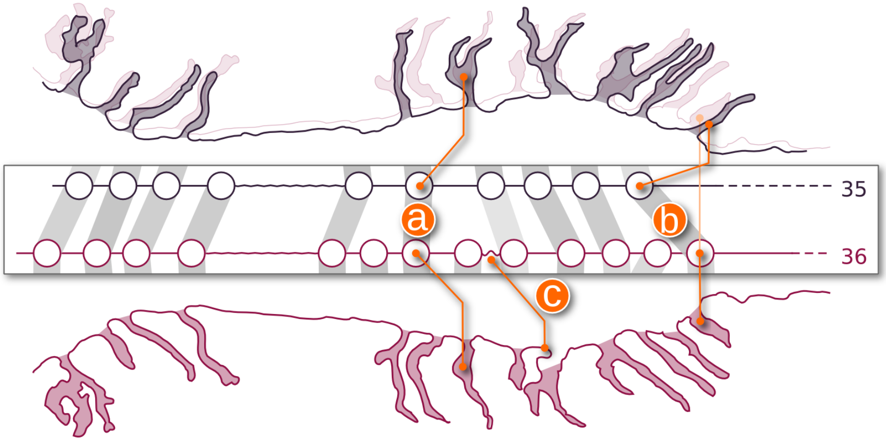
For the validation of the connection assignment and the following sections we use a subset of a wild-type image stack, which comprises 21 adjacent slices. The resulting surface map provides an overview of the fold traversal in the muscle membrane for a surface extent of about 7.5 µm × 1.4 µm.

To validate the connection method that links folds across image slices detailed in section 6.5.3 we investigated all folds manually. All existing connections were found correctly by the algorithm. This includes the detection of folds that develop secondary fold structures as shown in Figure 6.10 (labeled (a) in both cases). All erroneously connected folds were assigned low probability values (shown by the very light gray color of the connections).

Figure 6.10 gives two examples of computed matchings. In both cases we see an excerpt of the surface map comprising two slices (= images). Above the surface map, the corresponding visualization from the slice view is presented (pink and purple curves). To aid the validation, the pink curve is mirrored and displayed



(A)



(B)

FIGURE 6.10: Two extractions from the surface map annotated with the corresponding segmentations. The pink lower segmentation is a mirror image of the semi-transparent pink one on the top, in both cases.

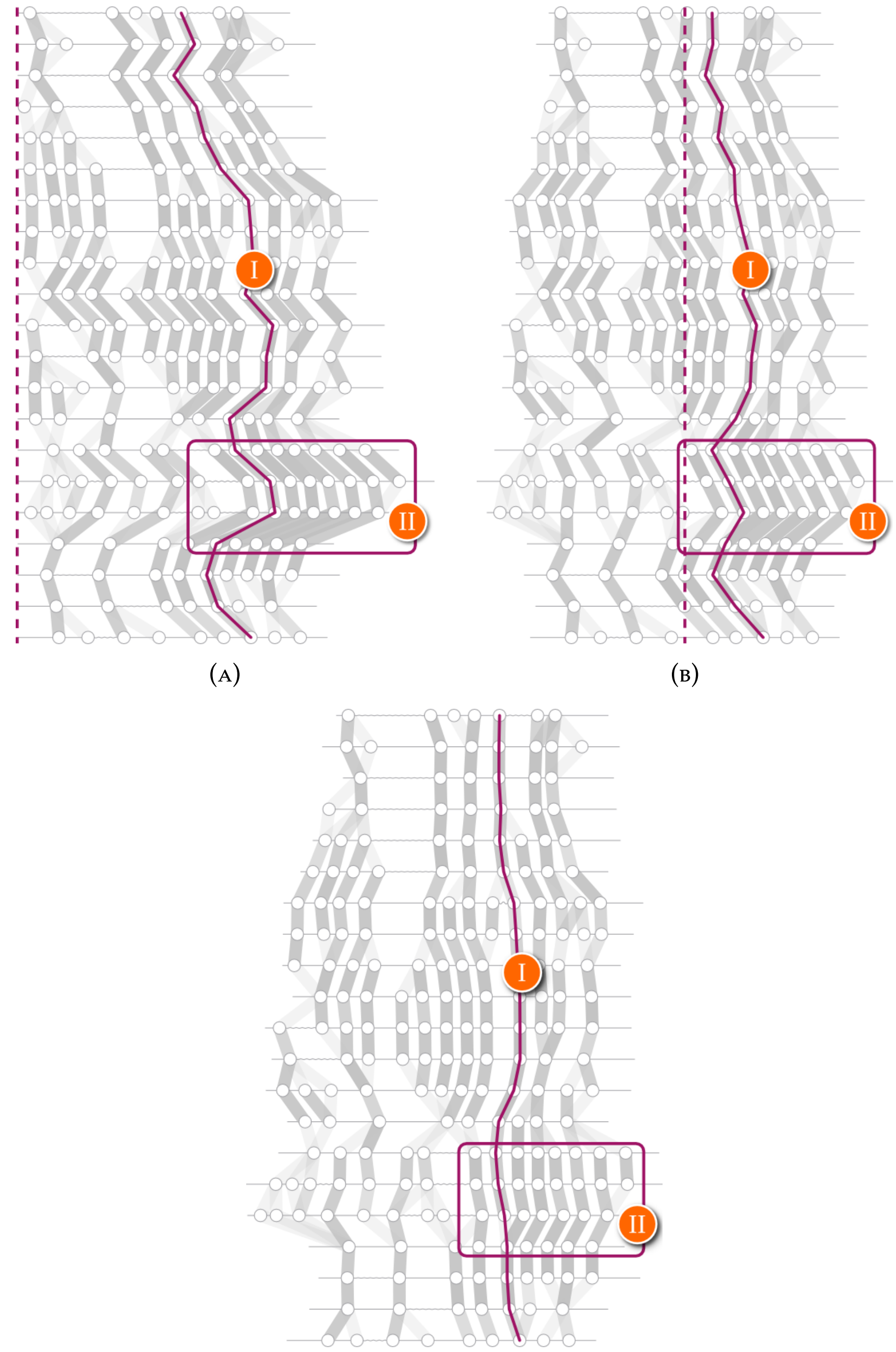
again below the corresponding surface map row. Case (A) contains mainly one-to-one pairings in full gray color. This means that the algorithm is highly confident about the matching, which can be visually validated. The part of the surface map annotated with (c) exemplifies a subsection of the segmentation with strong local "wiggling" which is also reflected in the surface map.

Case (B) is more complex with its appearance/disappearance of folds between the slices. Again (a) marks examples of regular matches. Also, (b) is a regular match which features large spatial distances. Judging only from the surface map this seems to be a linking error, but looking at the segmentation data, we can easily verify that the connection is correct. The misinterpretation is caused by spatial distortions in the surface map.

6.7.4 PRESERVATION OF SPATIAL FEATURES

A second aspect we analyzed in detail is the preservation of spatial features in the data such as clusterings of folds and sizes of gaps. Again we found overall a very good agreement and will give some evidence using the data depicted in Figure 6.10b. From the polyline data, we can observe that the folds cluster together in a group of four folds on the left and six/nine folds on the right-hand side. These clusters are preserved in the surface map. This also facilitates the identification of specific folds in the two views (surface map and slice view).

Furthermore, we observe that the tortuosity of edges in the surface map is an intuitive encoding of the membrane's wrinkling between two folds. It gets stronger as the Euclidean distance and the arc length of the segmented curve diverge. Both (c) labels in Figure 6.10 mark edges of particular tortuosity. Looking at the segmentation data, we can observe that this line type indicates parts of the segmentation that may be erroneous. The large distance between Euclidean distance and arc length either stems from a jagged segmentation (which is not commonly expected in the given data) or it may indicate that a fold was missed by the automatic detection. Concerning the case shown in Figure 6.10b, a fold next to the marked membrane part is indeed detected, but its boundaries are not correct. In case (A) (Figure 6.10a), however, a small fold was missed completely. Our recommendation for experts in the proofreading process is to check tortuous parts (which are usually rare) in the surface map for errors.



(a)

(b)

(c)

Figure 6.11: The effect of different horizontal alignment strategies of rows. (a) Default alignment. (b) Centered alignment. (c) Smart alignment. The positive effects – (I) edge straightening and (II) the avoidance of overplotting – of our alignment strategy (c) can, in contrast to (a) and (b), be observed particularly well in the lower parts.

6.7.5 Discussing the Inter-row Alignment

Figure 6.11 shows the impact that the kind of the horizontal alignment of rows has on the appearance of the surface map. As detailed before, the segmentation per image has no clearly-defined start and end point. We started originally with a simple left-aligned visualization as depicted in Figure 6.11a. This leads to strong artifacts in the surface map, such as sloping connections across slices that are not present in the microscopic images. Centering of the rows already yields a noticeable improvement of the fold alignment and eases readability. It reduces overplotting of the connection bars and facilitates the tracking of nodes between the rows. The red outline in Figure 6.11b, however, highlights an area that still suffers from strong distortions. However, the layout still suffers from strong distortions. Using our new alignment technique (Figure 6.11c) as proposed in Section 6.5.3, we can further increase visual consistency. This also improves readability in general as discussed in [84].

A common problem—independent of the used horizontal alignment method—is that potential shearing within the image stack is not taken into account. This problem already occurs in an earlier stage of the working pipeline, namely the SIFT-based [85] alignment of the microscopic images in the stack. Even if a potential shift was not considered during this process, none of the mentioned alignment methods would take such a drift into account. However, since our primary interest concerns topological structure and relative distances though, this is currently of less concern.

6.7.6 Evaluating the Quality Assessment

For the evaluation of the automatic quality assessment that we presented in Section 6.3, we performed a manual segmentation quality assessment to obtain ground truth. We asked a biologist who is an expert with microscopic images of NMJs to assess the segmentation quality of each of the 219 folds in our test data set. She rated the segmentations of each fold (which were shown to her without any quality information) by comparing them with the associated microscopic image data. She used a scale of three grades and the following criteria for segmentation quality. a *poor* segmentation lies partially on itself or takes severe short-cuts (instead of following the sarcolemma that is more invaginated towards the muscle tissue). *Medium* segmentations have parts that run alongside of the sarcolemma instead of on it and take short-cuts where the sarcolemma approaches itself. All other folds are rated *good*.

We discretized the quality values obtained by our automated estimates in the following way: *good* for values in $[1.0, 0.8)$, *medium* for values in $[0.8, 0.5)$, and *poor* for values in $[0.5, 0.0]$. For the evaluation of our quality assessment method we used the *polychoric correlation* [86], which is specifically geared towards measuring rater agreement on ordinal data. We obtained a correlation coefficient of 0.52, which denotes a positive statistical dependence of medium strength between the automatic and the manual assessment. Thus, the presented quality measure is a sound basis for modeling the segmentation quality.

6.7.7 Testing the NeuroMap Framework

For the evaluation of the proofreading and correction process we conducted an informal interview with a biologist, who tested NeuroMap on the described subset. After an introduction into the NeuroMap framework, we asked her to improve the quality of the data set, while explaining her strategies to us. She started selecting one of the five folds on the surface map that had the worst quality score and corrected it in the slice view. She used both modes (moving and erasing vertices) of the provided editing tool for improving the segmentation. After being satisfied with the enhancement, she went on to the next low-quality fold, until she had revised all folds of the lowest quality score. This procedure increased the automatic score of one fold by two levels, while another one was even improved by three levels. The remaining three folds retained their initial value. Regarding their corresponding microscopic image data, even for an expert it was very hard to recognize the sarcolemma. The expert stated, that she would like to be able to mark folds in the surface map that were already edited and approved. After having visited the five lowest-rated segmentations, she continued examining the folds of the second lowest quality level.

The evaluation process resulted in the following insight: the quality summary chart gives a first concise overview of the segmentation quality of entire slices. Having a look at this summary first, the expert can easily examine slices that are located at the ends of the quality spectrum. Thus, the experts get an idea of the quality spectrum of their data. We observed in our current scenario that the segmentation of the first slice in the stack (topmost slice of surface map in Figure 6.5, left) is of extraordinarily high quality. Consulting the source data, we found out that the respective microscopic image was in remarkably sharp focus, permitting high-quality segmentations. The expert stated that when she would have begun the proofreading process with the first slice (as conventionally), she would have been much too accurate when correcting, aiming for a quality level that just could not be retained in the other slices. The quality summary chart prevented her from spending too much time on comparatively high-quality data portions.

6.8 CONCLUSION

In this chapter, we presented a framework to support proofreading of surface segmentations from electron microscopic image stacks. For good overview and easy navigation within the data, we propose the surface map, an abstract 2D representation using key features of the surface as landmarks. We augment the visualizations with automatic error estimation. This gives users a quick overview of the amount and distributions of errors in their data. Integrated polyline correction mechanisms support users in the data cleaning process.

Using samples of images resulting from scanning electron microscopy that picture neuromuscular junctions, we could show the viability of our system. One open problem that we want to address in the future is a revision of the quality computation to take, e.g., low quality of the underlying image data into account. We also want to extend the surface representation to parametric surfaces for better curvature control. By all means, first examinations of the applicability of using splines for the sarcolemma representation yield promising results [87].

7 Parameter Space Analysis for Comparative Statistics

Topological and morphological descriptions of (sub-)cellular structures play a central role in the understanding of biological processes. Deriving such descriptions from image data, however, is a challenging task that has so far only been addressed for individual objects at a coarse resolution with small numbers of samples. For larger samples, the structured analysis is highly-challenging as little a priori knowledge exists.

In this chapter, we address the design of a generic parameter space for segmented objects that forms the basis for subsequent structural analysis. We will show how a structuring of the parameter space helps the user to identify the relevant configurations and to explore them in more detail. We present software that aids in the comparative parameter space analysis of tissue characteristics. Multiple views on the data are presented showing (i) enriched input image data, (ii) abstract representations of the parameter space, and (iii) depictions of individual parametric configurations.

Parts of this chapter have been published [11] and presented at the EuroVis Conference in 2013.

7.1 Introduction

Visual analysis of microscopy images has played a dominant role in the history of biology. For small samples and a limited number of images this has proven to be a very powerful analysis strategy. With the rise of high-resolution high-throughput microscopy, however, visual analysis is no longer feasible. Modern electron microscopes, for example, can automatically scan large tissue blocks resulting in image stacks consisting of several hundred images, each with a resolution of several gigapixels. Manual analysis in this setting is no longer practicable and the need for automatic support has been widely expressed in the past years [74].

Excellent progress has been achieved in the last years for images capturing cell data [68, 73], gene expression patterns [88, 89, 90], or cellular pathways [91, 92]. In these applications, there already exists a quite good understanding of how to quantify relevant parameters in the data using, for example, size, shape, or texture statistics which can be automatically computed using computer vision and machine learning algorithms.

For segmented image structures and their analysis, however, the required parameters are often not clear as there is yet too little understanding of what is in the data, how structures look like, and how much natural variability there is. A precise quantification has not yet been addressed as (a) the necessary image modalities to record such data have not been available and (b) tools that are currently available for the analysis of segmented image data are not yet powerful enough and often require a large amount of manual tweaking and implementation experience from the biologist.

The rapid acquisition of huge amounts of image data at a high level of detail provides unprecedented new insights and enables scientists to study the fundamental structure and organization of organisms on a subcellular level. Such data provide the basis for a comprehensive quantitative description of cells and their respective cell organelles. They allow for the quantification of individual structures using shape descriptors, as well as for the analysis of interrelationships using structural and topological descriptors. These descriptors hold the promise of telling what the "standard" cell looks like and how much natural variability there exists.

Regarding our application, a central question is whether or to what extent the fold structure influences biochemical processes at the synapses. Diseases such as muscular dystrophies and myasthenic syndromes are expected to correlate the morphology of neuromuscular junctions (NMJs). The coherency between the altered membrane's ultrastructure and the muscle function and disease symptoms needs to be explored. For this purpose, the differences in shape and topology of the junctional folds between a healthy and morbid muscle tissue and nervous system have to be studied. However, to derive viable results, not only differences of two exemplified healthy and morbid group members can be considered, but also intra-group within the sets, which need to be compared. Thus, an enormous amount of data sets and their intra- and inter-group structural discriminators have to be analyzed in a statistically meaningful manner. For this purpose, a highly specialized, however very flexible and adaptable, toolbox is needed.

7.2 The Parameter Space

The first step in the comprehensive quantitative description of biological structures is the definition of a parameter space that comprises all potentially relevant information concerning the segmented structures. Relevant parameters often comprise structural properties (e.g. length, width, volume, shape), textural properties (e.g. intensity, granularity, patterns), or neighborhood information (e.g. nearest object of same type, number of surrounding objects of different type). Commonly there is also secondary information such as the age of the subject the sample was taken from, the type of subject (e.g. healthy vs. pathological), or the location of the sample within a larger context (e.g. muscle tissue sample from the leg, heart, or face). The parameter space has to encompass all this information to permit a versatile subsequent analysis.

7.2.1 Biological Questions and Model Implications

Before going into detail about the precise definition of the parameter space, we want to look at typical questions that arise when analyzing segmented image data:

- In which features do healthy and pathological subjects differ?
- What is the mean shape and how does it vary?
- Does spatial location influence the segmented structures?
- How do structures change with increasing age?

These questions illustrate some of the fundamental difficulties in visualizing biological segmentation data, which strongly influence the theoretical data basis and the software design:

(i) **Problem specification:** The scientific question is commonly rather fuzzy. The transition from the general "I want to understand my data." to precise questions that can be answered using visualization or mathematical analysis is a central aspect in the "visualization" process.

(ii) **Feature selection and definition:** Features necessary to answer the extracted questions are commonly not clear a priori. A variety of potentially helpful features has to be identified, implemented, and tested.

(iii) **Interactive subset selection:** The software has to support database-like functionalities to select and/or group subsets of the data.

(iv) **Coupling of multiple algorithms interactively:** The users commonly require a coupled set of algorithms for a holistic analysis. Visualizations of the raw data with highlighting functionality have to be connected to parameter space visualization and exploration algorithms via linking-and-brushing.

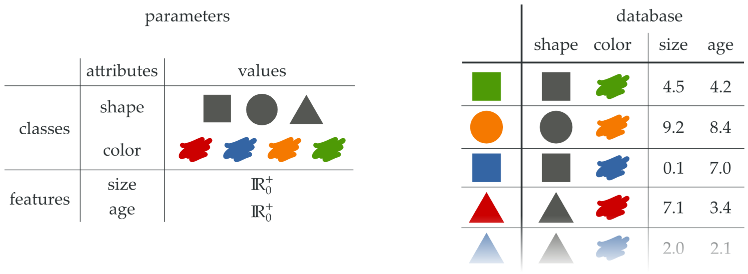
(v) **Ensemble visualization:** The software needs to be able to visualize ensembles to compare multiple subjects across different groups.

7.2.2 Features and Classes

An important part of the parametric data description is the distinction of two types of information: we will divide object parameters into features and classes. *Features* are quantifiable properties of segmented objects that are commonly directly measured using the input image/segmentation data. Examples are size, shape, or texture. *Classes* are properties of the segmented objects that are used to cluster data. Examples of classes are healthy vs. pathological, age group, species, or the location of a sample. During our research we identified some cases where classification was more difficult. These are commonly derived attributes such as spatial location within the data set, which might be used for data aggregation (→ class) as well as statistical analysis (→ feature). The implications for the software design will be discussed later. Overall, we found this distinction very helpful to provide a first structuring of the data, which is used to add levels of detail to the analysis process, and to make the biologists pose their research questions more precisely.

Class information is commonly used to separate the segmented data into groups that are to be compared. The easiest setting is to choose one primary class, e.g., a certain age group, and compare it to the rest of the data. Often a more detailed analysis is necessary to derive meaningful information. For example, subjects of a certain age group may only differ in healthy subjects but not in pathological ones.

Feature information is the one that is subsequently analyzed and used to derive model information. In manual analysis of the feature space, biologists commonly rely on low-dimensional standard techniques such as histograms, scatterplots or



(A) (B)

FIGURE 7.1: Example of data objects with nominal class parameters and two continuous feature parameters. (A) Existing parameters. (B) Parameter storage per object.

heat maps [74]. For many data attributes, the distinction between feature and class is readily given.

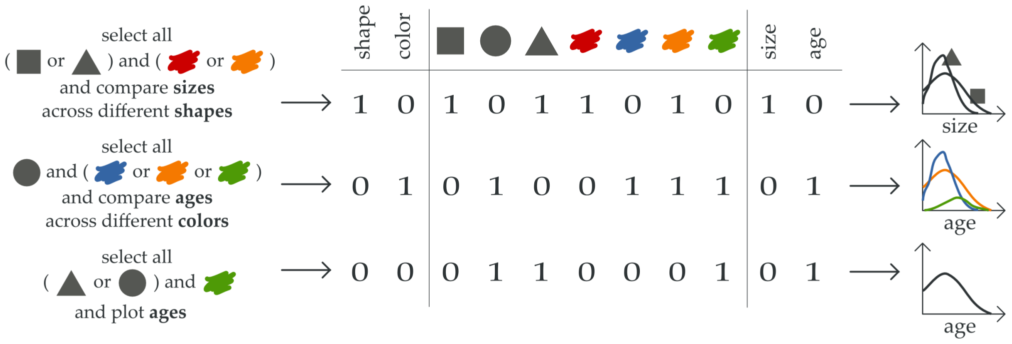
In summary, classes are for data selection and clustering and features are used to quantify commonalities and differences between subsets of the data.

7.2.3 THEORETICAL MODEL

Taking the previous considerations into account, the theoretical model for the parameter space of quantitative descriptors of segmented biological image data consists of two major parts for data classes and features respectively. The first n dimensions of the parameter space are dedicated to the n classes. In many cases classes the nominal values that have no implicit ordering, such as type of species, gender, or spatial location. There also exist classes with ordinal and even continuous properties, such as age or data acquisition parameters. The next m dimensions represent the derived object features. Often, these are continuous scalar quantities but they may also have more complex structure such as shape information or graph structures to represent neighborhood properties.

Data points are stored in the common form: each segmented object is assigned an $(n + m)$-dimensional vector containing class and feature information. Figure 7.1 shows a simple example with $n = 2$ and $m = 2$. In our collaborations, we often revised and extended the feature space, which does not affect the theoretical model, but has strong implications on the resulting implementation.

Operations on the data directly arise from the feature and class discussions. For classes, we require functionalities for interactive selection and grouping of data.



(A) (B) (C)

FIGURE 7.2: Three examples of exploratory queries. (A) Query in natural language. (B) Normalized binary representation of the query. (C) Answer of query as statistics view.

For features we need data analysis routines from statistics and machine learning to extract data characteristics as well as clusters and structures that emerge in the parameter space.

In Section 7.2.1 we collected some formulations of potential questions to the data set. Queries of this type can be described by a combination of parameters. Figure 7.2a shows such a query for the data parameters that are shown in Figure 7.1. For a dataset these queries can be transformed into a normalized binary representation (Figure 7.2b), by the selection (1) and deselection (0) of comparison parameters (shape, color), individual class values (rectangle, circle, triangle, red, blue, orange, green), and features (size, age). The result of the queries are statistical diagrams on the data portions of the desired feature (Figure 7.2c). The queries result in statistical diagrams of the data portions showing distributions of the desired feature (Figure 7.2c). The binarization of the parameters yields the key to the structured navigation of the parameter space.

For the visual inspection of the parameter space, information visualization techniques and statistical graphics are necessary to render the parameter space. Augmented image visualizations including highlighting and interactive selection are required for the image and segmentation data. As discussed in Section 7.2.1, all methods have to be connected via linking-and-brushing. Moreover, they are required to be interactive and to support ensemble visualization.

	parameters	values
classes	health	mdx | wt
	nearest tissue	glia | neuron | none
	type	primary[1] | secondary | internal
features	arc length	$\mathbb{R}^+$
	area	$\mathbb{R}^+$
	opening width	$\mathbb{R}^+$ | NaN
	spine length	$\mathbb{R}^+$ | NaN
	distance to next fold	$\mathbb{R}_0^+$

Table 7.1: The class and feature parameters of the fold dataset and their respective domains.

7.3 JUNCTIONAL FOLD DATA

The major goal of our software is to provide an intuitive interface for the comparative analysis of large parameter spaces. In particular, we concentrate on the exploration of features of junctional folds.

The underlying fold data set is obtained from the vectorized semi-automatically segmented micrographs of NMJs. For the enriched data view we build on the micrographs of neuromuscular junctions and pixel-based segmentations. The acquisition process of the microscopic images is described in detail in Section 2.2. The dataset of fold properties is gained from the pixel-based segmentations (see Section 4.1.1). The 3655 folds originate from 78 slices of six different synapses. The spacing between two slices amounts to approximately 0.7 µm. This distance results from a trade-off between the aspiration to build a dataset as large as possible and to guarantee a fair sampling of folds within the volume without redundancies.

The process of fold extraction is described in Section 5. The properties of the folds and their computation are detailed in Section 4.4. A fold inherits attributes from its ancestors in the hierarchical data model (see Section 4.3), e.g. *nearest tissue: glia* from its membrane or *health: mdx* from the synapse it belongs to. Table 7.1 summarizes the parameters of the fold dataset.

---

[1] In this application *fold type: single* is covered by *fold type: primary*.

7.4 TASKS

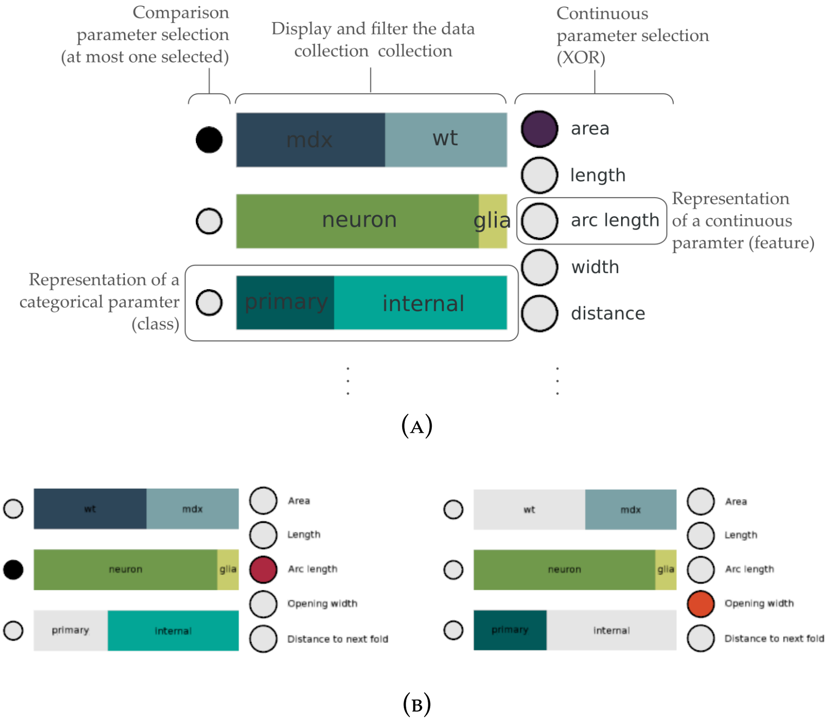
As geometric descriptions of cell tissue have so far only been analyzed with respect to very limited questions, it is not yet clear which parameters are going to be meaningful in answering more general questions, such as "What does a standard NMJ look like?". Hence, the **first task** consists of the development of a flexible data model that (i) can easily incorporate new classes and features, and (ii) that provides filtering and grouping mechanisms.

The **second task** consists of providing an overview of this inhomogeneous parameter space, which even for only few parameters consists of hundreds or even thousands of combinations, each of which may contain relevant information or not. With only three fold classes of size 2, 3, and 3, and five fold features, we obtain more than 2000 possible combinations that result in different pieces of information about the data. In Section 7.5 we will demonstrate our system with this parameter configuration. Adding additional fold characteristics results in an exponential increase of combinations, rendering the task of a systematic parameter space analysis very difficult. Hence, we need a visualization to structure and summarize the parameter space and its analysis.

Parameter space analysis provides a dual view on the data. Instead of drawing conclusions from the data about the parameter space, we analyze the parameter space to draw conclusions about the data. In order, to make these transitions comprehensible, a linked visualization of the different views on the data is required. The views have to support the following pieces of information: (i) presentation of the original segmented EM data, (ii) depiction of the parameter space and its structuring, (iii) summary of analyzed parameter space configurations, and (iv) comparative visualization of data attributes for a given set of data characteristics. **Task three** consists of the design and implementation of a software tool that supports these linked views.

7.5 VISUALIZATION

In this section we present the different parts of our visualization framework that we designed to meet the tasks that were formulated above. The interactive *navigation panel* and the *analysis record* for the visualization of the parameter space. A *statistics view* that shows the distribution of the feature values of the selected parameter



(A)

(B)

Figure 7.3: The navigation panel for the parameter space. (A) Functionality. (B) Potential parameter combinations represented by the navigation panel.

configuration. And an interactive *enriched data view* for displaying the segmentation and microscopic image data of individual data points.

7.5.1 Navigation Panel

The *navigation panel* summarizes the available parameters (classes and features) and serves as navigation interface for the parameter space. It is shown in Figure 7.3.

It is designed to give an overview over the dataset's composition and to navigate to a particular position in the parameter space. The navigation process will also be called *selection of a configuration* in the following.

The navigation consists of three units ordered in columns from left to right, which handle the following tasks:

1. select the data class used for comparison,
2. filter the data collection, and
3. select the required data features.

The second column represents the dataset according to the different data classes that are defined (we have these three parameters: *Synapse type* in shades of blue, *Cluster Type* in shades of green and *Fold Type* in shades of cyan). The dataset is represented using interactive bargrams as in [56]: Each colored and labeled rectangle behaves as a button, it can be selected or deselected. By default, all of them are selected. Every button represents a parameter value. The size of a button in a row corresponds to the percentage of the dataset that contains this parameter value. Thus, the user can keep a rough overview of the underlying data collection, while configuring the desired data filter. Not all theoretically possible combinations of selected and deselected data collection parts are sensible. For example, there is no reason to deselect representatives of all possible values for a parameter: this would results in selecting the empty set. Thus, the software prohibits the selection of such configurations.

The first column is tightly coupled with the second one. It specifies the data class that is used to divide the data into groups that are to be compared, for example, wild type mice vs. mdx mice. Exactly one of the data classes has to be selected and it requires at least two selected values within this data class. This condition is automatically satisfied by the software.

The third column shows which feature is currently selected for further examination. Further examination might be, e.g. regarding the frequency distribution. Each feature we defined for our data collection is listed here and can be selected. Again, not all combinations of data characteristics and attributes are possible, e.g., there exists no opening width for internal folds. Hence, the software automatically checks the validity of the input, prohibits inconsistent settings, and notifies the user.

We use consistent color coding within all views to provide a visual link between them.

7.5.2 Analysis Record

The *analysis record* is a visualization of the entire parameter space, which includes highlighting of already examined and/or favored parameter combinations. The user is able to access a particular position (configuration) in the parameter space by means of the introduced navigation panel.

We found that classical methods for visualizing parameter spaces such as scatterplot matrices or parallel coordinates do not suit our data, since our space is spanned by nominal data in few categories in all dimensions, i.e., data points are fairly sparse

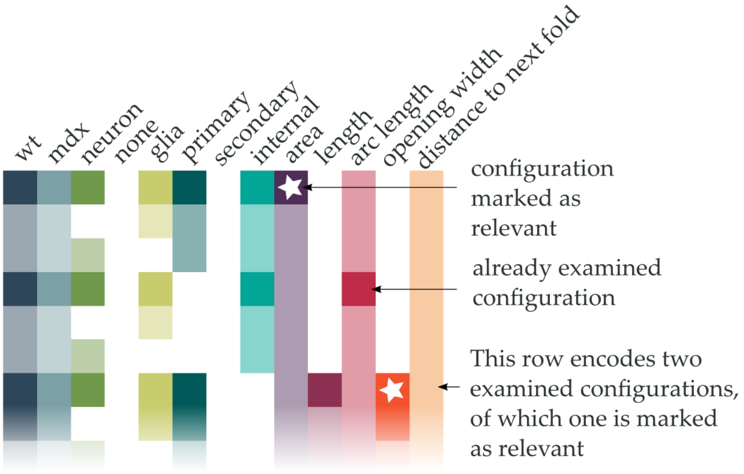


FIGURE 7.4: Part of the analysis record's list view (skipping the class values *none* and *secondary*). The cool colors on the left depict the parameters based on class values and the warm colors on the right those based on features. Stars indicate configurations marked by the user.

and do not form meaningful structures, and these representations are not intuitive. Formally the space is spanned using the power set minus the empty set of each class. For example, the class `nearest tissue` has three values and results in the following set of valid combinations in the parameter space *{{primary},{secondary}, {internal}, {primary, secondary}, {primary, internal}, {internal, secondary}, {primary, secondary, internal}}*. Composing this by the Cartesian product with the other classes, it is difficult for the user to keep track of the combinations. Furthermore a representation by parallel coordinates would suffer from severe occlusion, because the underlying data takes only a few discrete values.

Hence, we designed a list view as shown in Figure 7.4, that picks up the color scheme of the parameter space navigation panel as well as parts of its arrangement. The values are coded in a bit scheme that reflects the look of the navigation panel. Parameter space configurations that have already been visited are colored in full color, colors of possible configurations that have not yet been visited are attenuated.

A complete list of all configurations (positions in the parameter space PS) would have a length of

$$|PS| = m(n+1) \prod_{c \in \text{class}} (2^{|c|} - 1). \tag{7.1}$$

For our NMJ data example, this would result in $5 \cdot 4 \cdot (2^2 - 1) \cdot (2^3 - 1) \cdot (2^3 - 1) = 2940$ rows. For every newly added class this number will at least triple.

For compressing features that share the same combinations of a selected data class, values are collapsed. For each data class value combination (cool colors in the left half of the list view), we list all the respective features (warm colors in the right half) in the same row as shown in Figure 7.4. This shortens the list by a factor of at most m. Each row now represents up to m entries in the parameter space. Features that have been visited are selectively turned to full color.

The analysis record so far provides information on the visited and unvisited configuration in the parameter space. To serve as a digital record of which parameter combination has yet been examined, the user can mark highly interesting, relevant, or helpful configurations with a white star. Marking a configuration automatically moves the corresponding statistical plot to a separate view where they can be further compared and analyzed.

An additional option, complementing the parameter space selection described in Section 7.5.1, is the direct selection of parameter configurations via the analysis record, which provides a shortcut when the user identifies settings that have not yet been analyzed. By selecting an entry in the record, the corresponding statistical plot for this parameter setting is displayed.

Moreover, the analysis record offers an order of rows, according to the potential importance of the related statistics. An important parameter configuration might be one, that contains a comparative statistic plot with very different distributions. For example, a configuration where the distribution of fold lengths is very different for mdx or wt data might be considered important or interesting. Each statistics plot is rated according to the maximal pairwise Wasserstein distance—which is also known as earth mover's distance—between the distributions it contains. Consequently, configurations with a high rating are listed at the top of the analysis record and users can start their examination with potentially meaningful configurations.

7.5.3 STATISTICS VIEW

Each entry in the parameter space selects a subset of the data, determines a feature, and a comparison class. Thus, it might represent a query such as "Select all primal junctional folds that oppose a nerve cell and compare their lengths between wild-type and mdx mice." The resulting statistical information is presented in the statistics view (Figure 7.5).

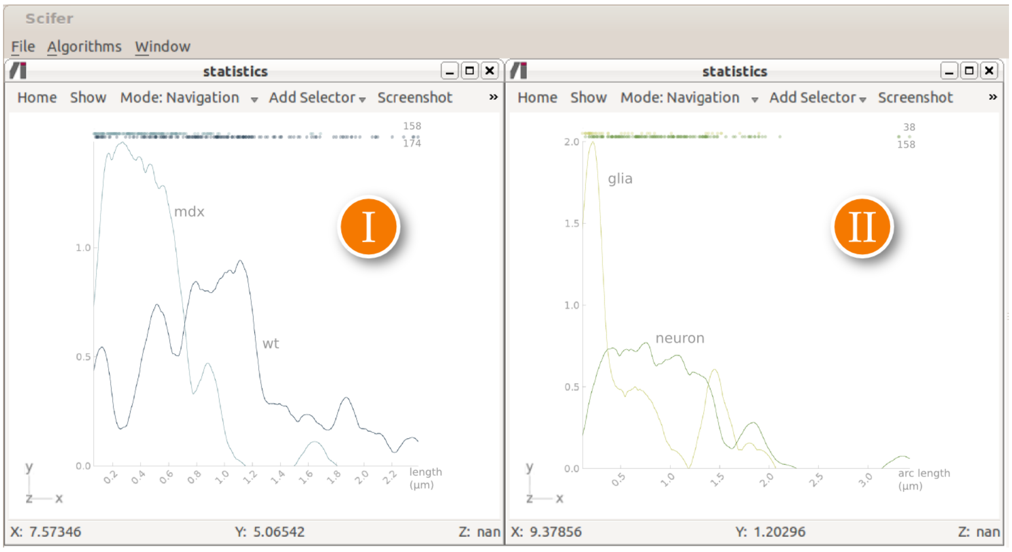
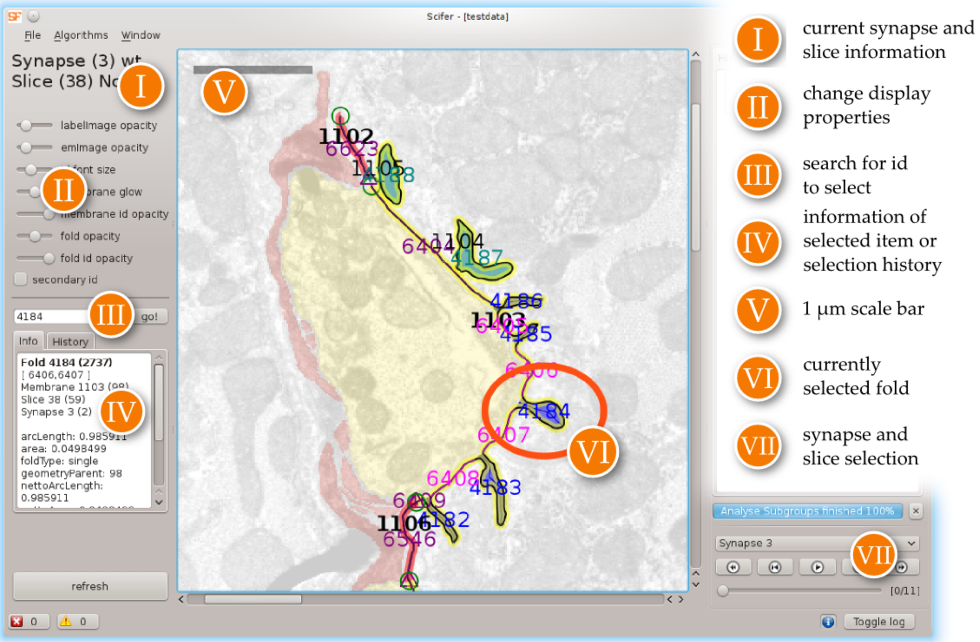


FIGURE 7.5: Two comparative statistics views. The color of the graphs refer to their corresponding class: (I) blue for comparison by `health` (*wt* vs. *mdx*) and (II) green for comparison by `nearest tissue` (*neuron* vs. *glia*). Each point at the top of the diagram is the interactive representation of a single fold.

Each diagram contains two or more frequency distributions, depending on how many class values are compared to each other. Here, we prefer a representation based on kernel density estimation (KDE) over histograms, because it supports the simultaneous display of distributions in one plot better. Histograms, at the other hand, suffer from occlusion when displaying different datasets at once. The diagram depicts the relative distribution of the parameter values, i.e., it does not matter if the compared data subsets differ in cardinality. The KDE diagram depends on the choice of a kernel function and the width of a smoothing window, the *bandwidth*. We use the Epanechnikov kernel and a bandwidth that depends on the data value distribution according to Silverman [93].

The statistical diagrams are complemented by interactive representations of the data, where each data object (here, a junctional fold) is represented by a data point. The points are rendered semi-transparently to account for varying numbers of points. By clicking on one point, the user gets information about the fold it is representing. The corresponding fold is also highlighted in the image data view that is introduced in the next section.



```
Fold 4184 (2737)
[ 6406,6407 ]
Membrane 1103 (98)
Slice 38 (59)
Synapse 3 (2)

arcLength: 0.985911
area: 0.0498499
foldType: single
geometryParent: 98
nettoArcLength:
0.985911
```

Figure 7.6: The enriched data view.

7.5.4 ENRICHED DATA VIEW

In the enriched image view (see Figure 7.6), the user gets a broad overview of the complete input data (VII). Here the original electron micrographs are displayed. The user can also choose to overlay the segmentation results with semitransparent label layers. The membrane is shown by its polyline representation and the segmented data. Extracted folds are highlighted with color and each is assigned an ID. Each information layer, image data, segmentation, extracted folds, and labels, can be hidden interactively via a control panel (II) to enable the user to focus on particular aspects.

The view is coupled with the data point plot of the statistics view: individual values can be selected there and their underlying fold is displayed and highlighted (VI) within its image and segmentation data context. In this manner, outliers or representatives with average feature manifestation and their properties (IV) can be displayed for further examination.

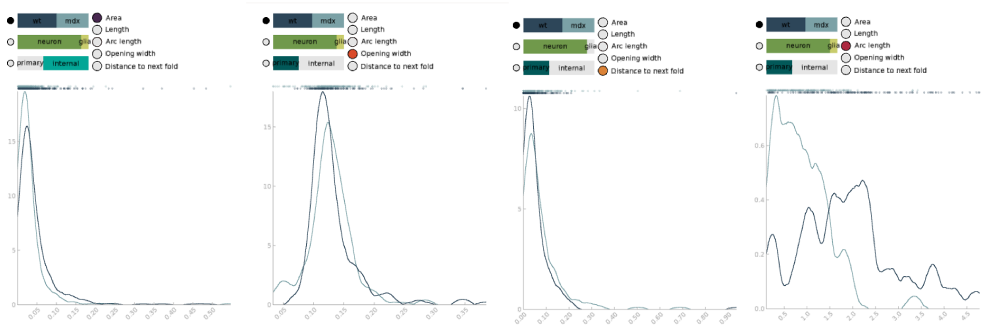


Figure 7.7: The statistical diagrams of the marked combinations are collected in a separate view, including a copy of the navigation panel showing the corresponding configuration.

7.6 RESULTS

The results consist of two parts. First, we will show findings that we made using the interactive interface, and second, we will detail the feedback that we received from our collaborators in biology.

7.6.1 PARAMETER SPACE ANALYSIS

The analysis session commonly starts with a random browsing of the parameter space, where different parameter combinations (also called configurations) are selected and further inspected. As detailed before, all visited configurations are logged in the analysis record, where relevant configurations can be highlighted.

As a first example, we compare wild-type and mdx synapses and can confirm that there are already visible differences in all data attributes though they are not very pronounced. For all data attributes, the distributions for wt synapses were more widely spread than those for mdx synapses. A subgroup analysis revealed that the joint analysis of internal and primal folds is not advisable, as parameters are quite different and distort the distributions. Deselecting internal folds as depicted in Figure 7.7 (right) revealed much more structure in the data. We still see that folds in the mdx mouse are in general shorter and that folds lengths in the wt mouse cluster into three heaps. The point representation at the top of the distribution allows us to inspect individual folds belonging to either of the three modes and get an intuition about what a typical fold looks like. Similar finding could be made for all other attributes.

The interactive linking between the statistical data representation and the image data view proved to be helpful to identify abnormal fold morphology. As shown in Figure 7.8 a peculiarity in the `area` attribute among all folds arises from an internal fold, that encloses neuron tissue. These result from the section and attest to the complex three dimensional structure of the synapse. Fold 287 could be easily identified as an outlier in the distribution which readily pops out in the statistical view. In this case, the reason for the appearance of an outlier is, that its exotic situation is not yet covered by the data model. Using the technique, e.g., looking for outliers, revealed several other unusual configuration such as extraordinarily long folds in mdx mice, which is very uncommon.

Checking the analysis record, we realized that we had only analyzed a very small subset of our parameter space and turned to alternative comparison modes. In the navigation panel, biologists were surprised about the large number of folds opposing glia cells. So far, biologists had assumed that folds only oppose neurons. With the interactive brushing&linking interface, we directly went to the corresponding folds in the image view. Changing the comparison criterion to this class of characteristics, enabled us to investigate differences between folds close to a glia cell and those close to a neuron.

During the analysis, the user can tag all configurations that they found interesting with a star. The tagging of promising configurations along with the display of their statistical diagrams in an overview as shown in fig. 7.7 was of great benefit for the subsequent summary of general findings and the identification of outliers.

7.6.2 User Feedback

In general, the biologists found the current interface very useful and intuitive. They noted that all four views are useful and particularly liked the easy navigation between image and parameter space. In the following we detail improvements that they suggested.

After having identified unusual folds or ones that contain errors, it would be helpful to eliminate them from further analysis to derive more robust results. In this context, they also asked for an automatic detection of anomalies, which is yet very difficult as so little is known about what is “normal”. A first approach in this direction would be to identify outliers in the statistical views and highlight corresponding parameter settings.

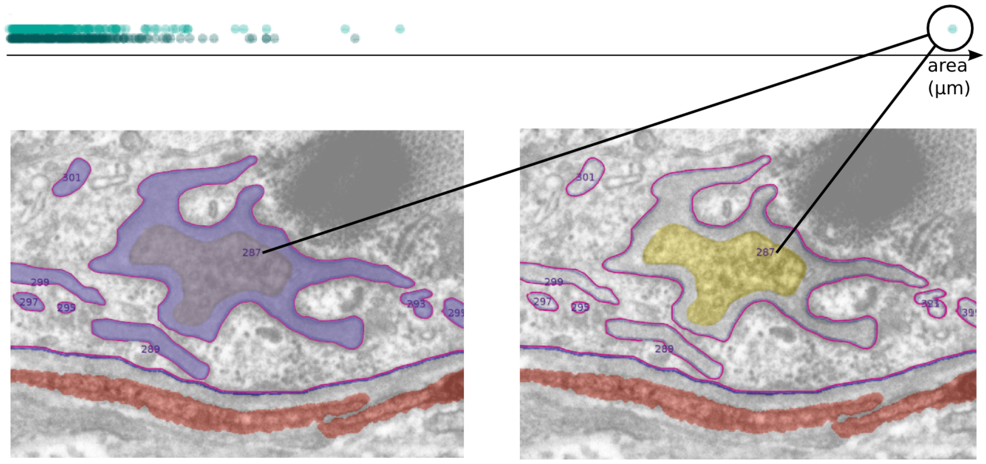


Figure 7.8: An outlier occurred in the statistics view concerning the *area* attribute. The data point referred to the corresponding fold in the image view. The extraordinary large size results from an internal fold that encloses neuron tissue. This case is not yet supported in the model.

So far the brushing option is linked to individual folds. Our users also asked for an extended brushing functionality that covers entire parameter ranges. Later we were able to meet this desire: In addition to the *enriched image view* we provide the *surface map* that is introduced as part of the NeuroMap framework in Section 6.5. The surface map provides a compact overview across multiple slices, where many folds can be highlighted and displayed simultaneously.

A third suggestion for improvement concerned additional attributes, where they would like to have a level of detail method included to not only compare characteristics of individual folds but cluster information of folds that are part of the same subsection of the membrane.

Additionally they identified unexpected structures that cannot be integrated in current data model, yet. Figure 7.8, for example, shows a muscle tissue that completely surrounds neuron tissue (concerning the section's surface). Such configuration are not yet covered by current parameters and require the further extension of the data model, and hence, the parameter space as well.

Regarding the enriched overview, it arose that encoding the affiliation of a fold or of a part of a membrane to either neuron, glia or intercellular space is not sufficient. Thus, we were asked to take the width of the synaptic cleft as an additional attribute into account and to provide a floating affiliation to neuron, glia or intercellular space.

In the future we will provide the functionality to remove individual e.g. unusual folds from analysis aiming for more robust results. The brushing&linking between the statical and the image view will be extended to cover parameter ranges and to highlight similar folds at once. Thus, in addition to the *enriched image view* we provide the *surface map* that is introduced as part of the NeuroMap framework in Section 6.5. The surface map provides a compact overview across multiple slices, where many folds can be highlighted and displayed simultaneously.

7.7 Conclusion

In this paper we described a theoretical model for the design of parameter spaces for the analysis of segmented structures in biological image data. We exemplified our considerations using examples from electron microscopy images of neuromuscular junctions in mice introduced a graphical interface to aid the examination of the parameter space for the analysis of descriptors for cell organelles. Here we presented an application on junctional folds.

For the classification of the fold dataset, we defined multiple classes and continuous features.

With our software we want to provide an intuitive overview over the dataset, allowing biologists to compare statistical data for multiple class parameter combinations. Thus we introduced four different interlinked kinds of views: The *enriched image view* shows the original dataset and the identified objects like folds and segmented cell organelles. This provides the user with the possibility to connect the abstract data representations from the other views with their underlying spatial information. The second view represents the parameter space and consists of the *navigation panel* allowing the selection of particular parameter configurations and the *analysis record* that logs the users' analysis history and allows them to mark relevant configurations. Moreover, the currently selected configuration is shown in the *statistics view*, allowing for a comparison regarding the chosen characteristics.

The application of our work allowed for data analysis and statistical insight on NMJ datasets. Our users found our software useful and intuitive, particularly appreciating the direct connection between the enriched images and the statistical representations.

8 CONCLUSION AND OUTLOOK

This chapter summarizes the project and its challenges. Furthermore, it highlights the contributions of the author. Finally, it gives an overview of further research topics that may emerge from this work.

8.1 PROJECT SUMMARY

The thesis is part of an interdisciplinary project with contributions from biology, image processing, and scientific visualization. The research interest from biology lies on the neuromuscular junction (NMJ), the contact area between a neuron and a muscle cell. In this area, the muscle cell membrane is a surface of complex shape: it features characteristic folds, which are of high interest for biological research. This interdisciplinary project aims at the examination the junctional folds on the basis of image stacks obtained by electron microscopy.

Correlations between local morphological properties of the muscle membrane and chemical processes at different regions of the NMJ are assumed and need to be explored further. Moreover, the coherency between the morphology of the junctional folds and the muscular diseases needs to be examined. Thus, variabilities in the morphological properties of the muscle membrane need to be quantified.

For addressing these questions, by means of quantitative analysis, high-quality segmentations of the microscopic data are necessary. Thus, our team developed a segmentation method, tailored for the segmentation of junctional folds. In combination with our segmentation proofreading framework *NeuroMap*, we developed a powerful tool for the creation of high-quality segmentations of filigree, complex surfaces.

A cardinal problem concerns the quantitative analysis of morphological properties of junctional folds: It is not known a priori which properties are to of importance for the analysis. Thus, we developed techniques to support the explorative parameter space analysis for biological image data.

These novel techniques offer new data for biologists to interpret in order to answer the research questions concerning the morphology of NMJs.

8.2 CONTRIBUTIONS OF THIS THESIS

In my work for this interdisciplinary project I developed novel problem-oriented interactive visualization techniques to aid the examination and segmentation of NMJs.

A central aspect of this research project is its need for reliable segmentation data for large amounts of microscopic image data. With our novel approach *NeuroMap*, the segmentation can now be performed semi-automatically by non-experts and subsequently be proofread by an expert. This process is greatly enhanced by visualization as we can reliably detect errors in the segmentation and point the expert to respective portions of the data. We couple automatic feature-based classification of segmented data with visual feedback to support the proofreading expert in rapid data correction. An abstract surface representation, the surface map, specially designed for intricate, folded surface structures aids in the visual communication of data quality and navigation in the raw data. The surface map and its interactive coupling with the original spatial data provides a template for biologists to organize complex three-dimensional data in a clear and intuitive way.

Another necessary step towards a comprehensive examination of NMJ data is the decomposition and structuring of segmentation results for further analysis. We present a data model for a structured description of all crucial parts of an NMJ, that also models the relations among its entities. For the decomposition of segmented curves from NMJ image data into smaller units, we introduce an algorithm based on the curve’s local orientation. In this way we were able to create six data sets permitting a quantitative analysis of the fold and muscle membrane properties of NMJs for the first time.

The subsequent research process from the biologist collaborators is largely based on exploratory data analysis and hypothesis-testing. This demands for a quick and easy access to the data, to directly check different hypotheses in preparation for further studies. We designed a generic parameter space for segmented structures from biological image data and introduced a graphical interface to aid its exploration, combining both, parameter selection and data representation. The analysis record recommends promising parameter combinations for examination and logs the users' exploration history.

The results obtained from the use of the tools we developed were used in subsequent analysis by the biology experts.

8.3 OUTLOOK

Currently, our collaborations from biology are working on publications on the analysis of the junctional folds. For this purpose, they use the data we extracted from their initial segmentations and the analysis tools we developed.

In particular, the surface map metaphor that we introduced in Chapter 6 turned out to yield fruitful results. We recall that we described the migration of AChRs in the introduction in Chapter 1. They migrate over time from the periphery of an NMJ to its center. Given these migration data, these movements can be visualized using the surface map as a reference frame. Simultaneously, the surface map may encode fold properties by color as it does for the segmentation quality in the NeuroMap framework.

From the sub-projects that were introduced in this thesis arise several possible starting points for further work on this interdisciplinary project.

The proofreading process of segmentations of the sarcolemma does not yet consider internal folds. They are not integrated into the abstract representation of the sarcolemma, the surface map. However, the internal folds are crucial for a correct 3D reconstruction of the sarcolemma. For their integration into the surface map, several questions have to be considered for them, such as an appropriate spatial position in the surface map and the determination of correspondences to folds of adjacent slices.

Another open task concerns the utilization of the Livewire segmentation technique in the curve editing process of the segmentation correction. As an additional option for the correction procedure, it would speed up the entire process even more.

Acronyms

AChR	nicotinic acetylcholine receptor 4, 5, 13, 105
mdx	X chromosome-linked muscular dystrophy 16, 18, 35
NMJ	neuromuscular junction 1, 3–8, 13–15, 18–21, 29, 30, 37, 41, 46, 59, 60, 86, 91, 92, 95, 102–105
SEM	scanning electron microscope 17, 19, 20
SEM	scanning electron microscopy 17, 19, 20
TEM	transmission electron microscope 17, 19
TEM	transmission electron microscopy 17, 19
wt	wild-type 17, 18, 35

GLOSSARY

Dmd$^{\mathrm{mdx}}$ Name for a mouse strain where the Dmd gene carries an mdx mutation. 16, 29

dystrophin A protein that is present in the sarcolemma. A lack of dystrophin causes muscular dystrophies. 16

glia Non-neuron cells in the nervous tissue. They provide support, sheath and electric isolation for neurons. 13

Livewire A segmentation technique for selecting regions in an image by a series of fix points on the regions boundary. 65, 77, 78

motor end plate Area on the surface of a muscle fiber where a neuron interacts with the fiber via the neuron's synaptic bouton. 13

muscle fiber A muscle cell of the striated muscle tissue. 9, 109

myocyte A muscle cell. 9, 11

sarcolemma The cell membrane of a muscle fiber. 11, 15, 109

segment A region of an image that results from a segmentation. E.g. a single cell in microscopic images of tissue samples. 21

segmentation Masking of similar regions in image data. E.g. single cells in microscopic images of tissue samples. 21, 109

synaptic bouton (also pre-synaptic bouton) Part of a neuron located at the end of the axon and interface for the interaction with other cells. 11, 109

wild-type Label for healthy lab mouse. 17, 29, 35, 78, 107

SYMBOLS

$\mathscr{C}$	Curve
$\mathcal{L}$	Arc length
NaN	Not a number
$\mathcal{L}_k$	Partial arc length
$\mathcal{P}_\circ$	Polygon
$\mathcal{P}$	Polyline
$\mathbf{s}_i$	The ith segment of a polyline/polygon
$\mathbf{v}_i$	The ith vertex of a polyline/polygon

BIBLIOGRAPHY

[1] M. Akaaboune, S. M. Culican, S. G. Turney, and J. W. Lichtman. “Rapid and Reversible Effects of Activity on Acetylcholine Receptor Density at the Neuromuscular Junction in Vivo”. In: *Science* 286.5439 (1999), pp. 503–507. DOI: `10.1126/science.286.5439.503`.

[2] T. M. J. Salpeter, M. M. Salpeter, E. E. Sejnowski, T. J. Stiles, and J. R. Bartol. “New Insights from Reconstructions and Monte Carlo Simulations with MCell”. In: *Synapses*. Ed. by W. M. Cowan, T. C. Suedhof, and C. F. Stevens. The Johns Hopkins University Press, Jan. 2000. Chap. Synaptic Variability, pp. 681–731.

[3] J. R. Stiles, W. C. Ford, J. M. Pattillo, T. E. Deerinck, M. H. Ellisman, T. M. Bartol, and T. J. Sejnowski. “Spatially realistic computational physiology: Past, present and future”. In: *Parallel ComputingSoftware Technology, Algorithms, Architectures and Applications*. Ed. by G. Joubert, W. Nagel, F. Peters, and W. Walter. Vol. 13. North–Holland, 2004, pp. 685–694. DOI: `10.1016/S0927-5452(04)80085-6`.

[4] K. Tai, S. D. Bond, H. R. MacMillan, N. A. Baker, M. J. Holst, and J. A. Mccammon. “Finite Element Simulations of Acetylcholine Diffusion in Neuromuscular Junctions”. In: *Biophysical Journal* 84 (2003), pp. 2234–2241.

[5] I. Röder, K. Choi, M. Reischl, Y. Petersen, M. Diefenbacher, M. Zaccolo, T. Pozzan, and R. Rudolf. “Myosin va cooperates with PKA RIα to mediate maintenance of the endplate in vivo”. In: *Proceedings of the National Academy of Sciences of the United States of America (PNAS)* 107.5 (2010), pp. 2031–2036.

[6] I. V. Röder, Y. Petersen, K. R. Choi, V. Witzemann, J. A. Hammer III, and R. Rudolf. “Role of Myosin Va in the Plasticity of the Vertebrate Neuromuscular Junction *in Vivo*”. In: *PLoS ONE* 3.12 (Dec. 2008), e3871. DOI: `10.1371/journal.pone.0003871`.

[7] J. Stegmaier, N. Peter, J. Portl, I. V. Mang, R. R. Schröder, H. Leitte, R. Mikut, and M. Reischl. "A Framework for Feedback-based Segmentation of 3D-Image Stacks". In: *Current Directions in Biomedical Engineering* 2.1 (Sept. 2016), pp. 437–441. DOI: `10.1515/cdbme-2016-0097`.

[8] J. Portl, J. Stegmaier, I. V. Mang, R. R. Schröder, M. Reischl, and H. Leitte. "NeuroMap: Support for the Proofreading and Correction of Surface Segmentations in Image Stacks". In: *Computer Graphics Forum* (2016). Under Review.

[9] J. Portl, M. Reischl, J. Stegmaier, R. R. Schröder, I. V. Mang, and H. Leitte. "Visualization for Error-controlled Surface Reconstruction from Large Electron Microscopy Image Stacks". In: *IEEE Vis: Visualization in Practice*. 2015. URL: `http://julia.portl.space/Papers/portl15a.pdf`.

[10] J. Portl and H. Leitte. *Fold detection in cell membrane segmentations of neuromuscular junctions*. Tech. rep. Revised Version from 2016. 2013. URL: `http://arxiv.org/pdf/1605.09250.pdf`.

[11] H. Leitte, J. Portl, I. V. Röder, R. R. Schröder, and I. Wacker. "Towards a Structured Analysis of Quantitative Descriptors from Segmented Biological Image Data". In: *Visualization in Medicine and Life Sciences*. Ed. by L. Linsen, H.-C. Hege, and B. Hamann. The Eurographics Association, 2013. DOI: `10.2312/PE.VMLS.VMLS2013.025-029`.

[12] M. P. McKinley and V. D. O'Loughlin. *Human Anatomy*. McGraw-Hill Higher Education, 2005.

[13] E. C. Klatt. *Normal Histology*. The Internet Pathology Laboratory for Medical Education. Distributed by the University of Utah Eccles Health Sciences Library. 2016. URL: `http://library.med.utah.edu/WebPath/jpeg2/MUSC012.jpg`.

[14] R. A. Bergman, A. K. Afifi, and J. Paul M. Heidger. "Atlas of Microscopic Anatomy". In: *Anatomy Atlases* (2016). URL: `http://www.anatomyatlases.org/`.

[15] J. Gilbert. *The Human Nervous System*. BiologyMad. URL: `http://www.biologymad.com/master.html?http://www.biologymad.com/asbiology.htm`.

[16] D. S. Auld and R. Robitaille. "Perisynaptic Schwann cells at the neuromuscular junction: nerve-and activity-dependent contributions to synaptic efficacy, plasticity, and reinnervation". In: *The Neuroscientist* 9.2 (2003), pp. 144–157.

[17] R. J. Balice-Gordon. “Schwann cells: Dynamic roles at the neuromuscular junction”. In: *Current Biology* 6.9 (1996), pp. 1054–1056. DOI: `10.1016/S0960-9822(02)70662-3`.

[18] H. Darabid, A. P. Perez-Gonzalez, and R. Robitaille. “Neuromuscular synaptogenesis: coordinating partners with multiple functions”. In: *Nature Reviews Neuroscience* 15 (Nov. 2014), 703–718. DOI: `10.1038/nrn3821`.

[19] Sample preparation, photography, microscopy, segmentation by Ira V. Mang, Cryo-Electron Microscopy Group, BioQuant, Heidelberg University.

[20] J. Ude and M. Koch. *Die Zelle: Atlas der Ultrastruktur*. G. Fischer, 1994.

[21] S. I. Fox. *Human Physiology*. McGraw Hill Higher Education, 2015.

[22] G. Bulfield, W. G. Siller, P. A. Wight, and K. J. Moore. “X chromosome-linked muscular dystrophy (mdx) in the mouse”. In: *Proceedings of the National Academy of Sciences of the United States of America (PNAS)* 81.4 (Feb. 1984), pp. 1189–1192.

[23] D. J. Blake, A. Weir, S. E. Newey, and K. E. Davies. “Function and genetics of dystrophin and dystrophin-related proteins in muscle”. In: *Physiological Reviews* 82.2 (2002), pp. 291–329.

[24] T. Liu, E. Jurrus, M. Seyedhosseini, M. Ellisman, and T. Tasdizen. “Watershed Merge Tree Classification for Electron Microscopy Image Segmentation”. In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR)*. 2012, pp. 133–137. URL: `http://www.sci.utah.edu/publications/liu12/Liu_ICPR2012.pdf`.

[25] W. Khan. “Image Segmentation Techniques: A Survey”. In: *Journal of Image and Graphics* 1.4 (2013), pp. 166–170.

[26] C. Sommer, C. Straehle, U. Koethe, and F. A. Hamprecht. “ilastik: Interactive Learning and Segmentation Toolkit”. In: *8th IEEE International Symposium on Biomedical Imaging (ISBI 2011)*. 2011.

[27] D. Stalling, M. Westerhoff, and H.-C. Hege. “amira: A Highly Interactive System for Visual Data Analysis”. In: *Visualization Handbook*. Ed. by C. D. Hansen and C. R. Johnson. Butterworth-Heinemann, 2005, pp. 749–767. DOI: `10.1016/B978-012387582-2/50040-X`.

[28] A. K. Al-Awami, J. Beyer, H. Strobelt, N. Kasthuri, J. W. Lichtman, H. Pfister, and M. Hadwiger. “NeuroLines: A Subway Map Metaphor for Visualizing

Nanoscale Neuronal Connectivity". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2369–2378.

[29] D. Meyers, S. Skinner, and K. Sloan. "Surfaces from Contours". In: *ACM Transactions on Graphics* 11.3 (July 1992), pp. 228–258. DOI: `10.1145/130881.131213`.

[30] M. Helmstaedter, K. L. Briggman, and W. Denk. "High-accuracy neurite reconstruction for high-throughput neuroanatomy". In: *Nature Neuroscience* 14.8 (Aug. 2011), pp. 1081–1088. DOI: `10.1038/nn.2868`.

[31] R. Sicat, M. Hadwiger, and N. J. Mitra. "Graph Abstraction for Simplified Proofreading of Slice-based Volume Segmentation". In: *Eurographics 2013 - Short Papers*. The Eurographics Association, 2013. DOI: `10.2312/conf/EG2013/short/077-080`.

[32] D. Haehn, S. Knowles-Barley, M. Robertsa, J. Beyer, N. Kasthuri, J. W. Lichtman, and H. Pfister. "Design and Evaluation of Interactive Proofreading Tools for Connectomics". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (Dec. 2014), pp. 2466–2475.

[33] A. K. Al-Awami, J. Beyer, D. Haehn, N. Kasthuri, J. W. Lichtman, H. Pfister, and M. Hadwiger. "NeuroBlocks–Visual Tracking of Segmentation and Proofreading for Large Connectomics Projects". In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 738–746.

[34] R. Maini and H. Aggarwal. "Study and comparison of various image edge detection techniques". In: *International Journa of Image Processing* 3.1 (2009), pp. 1–11.

[35] S. Lakshmi and D. V. Sankaranarayanan. "A study of Edge Detection Techniques for Segmentation Computing Approaches". In: *International Journal of Computer Applications* 1 (2010), pp. 35–41. DOI: `10.5120/993-25`.

[36] R. Muthukrishnan and M. Radha. "Edge detection techniques for image segmentation". In: *International Journal of Computer Science & Information Technology* 3.6 (2011), p. 259.

[37] E. Mortensen, B. Morse, W. Barrett, and J. Udupa. "Adaptive boundary detection using 'live-wire' two-dimensional dynamic programming". In: *Proceedings of Computers in Cardiology*. IEEE Computer Society Press, 1992, pp. 635–638.

[38] L. Antiga. "Generalizing vesselness with respect to dimensionality and shape". In: *The Insight Journal* 3 (2007).

[39] H. Zhang, J. E. Fritts, and S. A. Goldman. “Image segmentation evaluation: A survey of unsupervised methods”. In: *Computer Vision and Image Understanding* 110.2 (2008), pp. 260–280. DOI: `10.1016/j.cviu.2007.08.003`.

[40] D. R. Martin, C. C. Fowlkes, and J. Malik. “Learning to detect natural image boundaries using local brightness, color, and texture cues”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.5 (2004), pp. 530–549.

[41] F. Heckel, J. H. Moltz, H. Meine, B. Geisler, A. Kießling, M. D’Anastasi, D. Pinto dos Santos, A. J. Theruvath, and H. K. Hahn. “On the Evaluation of Segmentation Editing Tools”. In: *SPIE Journal of Medical Imaging* 1.3 (2014), 034005:1–034005:16. DOI: `10.1117/1.JMI.1.3.034005`.

[42] S. Knowles-Barley, M. Roberts, N. Kasthuri, D. Lee, H. Pfister, and J. W. Lichtman. “Mojo 2.0: Connectome annotation tool”. In: *Frontiers in Neuroinformatics* 60 (2013).

[43] F. Heckel, S. Braunewell, G. Soza, C. Tietjen, and H. K. Hahn. “Sketch-based Image-independent Editing of 3D Tumor Segmentations using Variational Interpolation”. In: *Proceedings of Eurographics Workshop on Visual Computing for Biology and Medicine*. The Eurographics Association, 2012, pp. 73–80. DOI: `10.2312/VCBM/VCBM12/073-080`.

[44] F. Heckel, J. H. Moltz, C. Tietjen, and H. K. Hahn. “Sketch-Based Editing Tools for Tumour Segmentation in 3D Medical Images”. In: *Computer Graphics Forum* 32.8 (2013), pp. 144–157. DOI: `10.1111/cgf.12193`.

[45] A. Karimov, G. Mistelbauer, T. Auzinger, and S. Bruckner. “Guided Volume Editing based on Histogram Dissimilarity”. In: *Computer Graphics Forum* 34.3 (2015), pp. 91–100.

[46] C. Grimm and M. Ayers. “A framework for synchronized editing of multiple curve representations”. In: *Computer Graphics Forum*. Vol. 17. 3. 1998, pp. 31–40.

[47] C. C. L. Wang and K. Tang. “Optimal boundary triangulations of an interpolating ruled surface”. In: *Journal of Computing and Information Science in Engineering* 5.4 (2005), pp. 291–301. DOI: `10.1115/1.2052850`.

[48] A. Inselberg. “The plane with parallel coordinates”. In: *The Visual Computer* 1.2 (1985), pp. 69–91.

[49] I. Borg and P. Groenen. *Modern Multidimensional Scaling: Theory and applications*. 2nd ed. Springer, 2005.

[50] S. Feiner and C. Beshers. “Worlds within Worlds : Metaphors for Exploring n-Dimensional Virtual Worlds”. In: *Proceedings of the 3rd Annual ACM SIGGRAPH Symposium on User Interface Software and Technology*. ACM Press, 1990, pp. 76–83.

[51] J. J. van Wijk and R. van Liere. “HyperSlice: visualization of scalar functions of many variables”. In: *Proceedings of the 4th Conference on Visualization*. San Jose, California: IEEE Computer Society, 1993, pp. 119–125. DOI: `10.1109/VISUAL.1993.398859`.

[52] S. Jayaraman and C. North. “A radial focus+context visualization for multidimensional functions”. In: *Proceedings of the Conference on Visualization*. IEEE Computer Society Press, 2002, pp. 443–450.

[53] Z. Guo, M. O. Ward, and E. A. Rundensteiner. “Model space visualization for multivariate linear trend discovery”. In: *IEEE Symposium on Visual Analytics Science and Technology (VAST)*. IEEE Computer Society Press, 2009, pp. 75–82.

[54] L. Tweedie, B. Spence, H. Dawkes, and H. Su. “The influence explorer”. In: *Conference Companion on Human Factors in Computing Systems*. CHI ’95. ACM Press, 1995, pp. 129–130. DOI: `10.1145/223355.223464`.

[55] R. Spence and L. Tweedie. “The Attribute Explorer: information synthesis via exploration. Interacting with”. In: *Computers* 11 (1998), pp. 137–146.

[56] K. Wittenburg, T. Lanning, M. Heinrichs, and M. Stanton. “Parallel bargrams for consumer-based information exploration and choice”. In: *Proceedings of the 14th Annual ACM symposium on User Interface Software and Technology*. Orlando, Florida: ACM Press, 2001, pp. 51–60. DOI: `10.1145/502348.502357`.

[57] D. Asimov. “The Grand Tour: A tool for viewing multidimensional data”. In: *SIAM Journal on Scientific and Statistical Computing* 6.1 (1985), pp. 128–143. DOI: `10.1137/0906011`.

[58] N. Elmqvist, P. Dragicevic, and J.-D. Fekete. “Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation”. In: *IEEE Transactions on Visualization and Computer Graphics* 14.6 (2008), pp. 1539–1148.

[59] J. E. Nam and K. Mueller. “TripAdvisor$^{\text{N-D}}$: A Tourism-Inspired High-Dimensional Space Exploration Framework with Overview and Detail”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.2 (2013), pp. 291–305. DOI: `10.1109/TVCG.2012.65`.

[60] S. van den Elzen and J. J. van Wvijk. “Small Multiples, Large Singles: A New Approach for Visual Data Exploration”. In: *Computer Graphics Forum* (2013). DOI: `10.1111/cgf.12106`.

[61] W. Javed and N. Elmqvist. “ExPlates: Spatializing Interactive Analysis to Scaffold Visual Exploration”. In: *Computer Graphics Forum* 32.3 (2013), pp. 441–450. DOI: `10.1111/cgf.12131`.

[62] A. Lex, N. Gehlenborg, H. Strobelt, R. Vuillemot, and H. Pfister. “UpSet: Visualization of Intersecting Sets”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1983–1992. DOI: `10.1109/TVCG.2014.2346248`.

[63] H. Piringer, W. Berger, and J. Krasser. “HyperMoVal: Interactive Visual Validation of Regression Models for Real-Time Simulation”. In: *Computer Graphics Forum* 29.3 (2010), pp. 983–992. DOI: `10.1111/j.1467-8659.2009.01684.x`.

[64] W. Berger, H. Piringer, P. Filzmoser, and M. E. Gröller. “Uncertainty-Aware Exploration of Continuous Parameter Spaces Using Multivariate Prediction”. In: *Computer Graphics Forum* 30.3 (June 2011), pp. 911–920.

[65] M. Booshehrian, T. Möller, R. M. Peterman, and T. Munzner. “Vismon: Facilitating Analysis of Trade-Offs, Uncertainty, and Sensitivity In Fisheries Management Decision Making”. In: *Computer Graphics Forum* 31.3 (June 2012), pp. 1235–1244. DOI: `10.1111/j.1467-8659.2012.03116.x`.

[66] T. Torsney-Weir, A. Saad, T. Möller, H.-C. Hege, B. Weber, and J.-M. Verbavatz. “Tuner: Principled Parameter Finding for Image Segmentation Algorithms Using Visual Response Surface Exploration”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (Dec. 2011), pp. 1892–1901. DOI: `10.1109/TVCG.2011.248`.

[67] A. J. Pretorius, M.-A. Bray, A. E. Carpenter, and R. A. Ruddle. “Visualization of Parameter Space for Image Analysis.” In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2402–2411.

[68] A. E. Carpenter, T. R. Jones, M. R. Lamprecht, C. Clarke, I. H. Kang, O. Friman, D. A. Guertin, J. H. Chang, R. A. Lindquist, J. Moffat, P. Golland, and D. M. Sabatini. “CellProfiler: image analysis software for identifying and quantifying cell phenotypes”. In: *Genome Biology* 7.10 (2006), R100.

[69] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. “Design Galleries: A General Approach to Setting Parameters for Computer

Graphics and Animation". In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. 1997, pp. 389–400.

[70] S. Bruckner and T. Möller. "Result-Driven Exploration of Simulation Parameter Spaces for Visual Effects Design". In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), pp. 1468–1476. DOI: `10.1109/TVCG.2010.190`.

[71] K.-L. Ma. "Image graphs – a novel approach to visual data exploration". In: *Proceedings of the Conference on Visualization*. VIS '99. San Francisco, California, United States: IEEE Computer Society Press, 1999, pp. 81–88.

[72] T. J. Jankun-Kelly and K.-L. Ma. "Visualization Exploration and Encapsulation via a Spreadsheet-Like Interface". In: *IEEE Transactions on Visualization and Computer Graphics* 7.3 (July 2001), pp. 275–287. DOI: `10.1109/2945.942695`.

[73] T. R. Jones, I. H. Kang, D. B. Wheeler, R. A. Lindquist, A. Papallo, D. M. Sabatini, P. Golland, and A. E. Carpenter. "CellProfiler Analyst: data exploration and analysis software for complex image-based screens". In: *BMC Bioinformatics* 9 (2008). DOI: `10.1186/1471-2105-9-482`.

[74] T. Walter, D. W. Shattuck, R. Baldock, M. E. Bastin, A. E. Carpenter, S. Duce, J. Ellenberg, A. Fraser, N. Hamilton, S. Pieper, M. A. Ragan, J. E. Schneider, P. Tomancak, and J.-K. Hériché. "Visualization of image data from cells to organisms". In: *Nature Methods* 7 (2010), S26–S41. DOI: `10.1038/nmeth.1431`.

[75] P. Selinger. *Potrace: a polygon-based tracing algorithm*. Tech. rep. http://potrace.sourceforge.net, 2003.

[76] S. Fortune. "Voronoi diagrams and Delaunay triangulations". In: *Handbook of Discrete and Computational Geometry*. Ed. by J. E. Goodman and J. O'Rourke. 2nd ed. Boca Raton, FL, USA: Chapman & Hall/CRC, 2004. Chap. 23, pp. 513–528.

[77] E. W. Dijkstra. "A Note on Two Problems in Connexion with Graphs". In: *Numerische Mathematik* 1.1 (1959), pp. 269–271.

[78] S. D. Cohen and L. J. Guibas. "Partial Matching of Planar Polylines Under Similarity Transformations". In: *Proceedings of the 8th Annual ACM–SIAM Symposium on Discrete Algorithms*. 1997, pp. 777–786.

[79] M. Neubert, H. Herold, and G. Meinel. "Assessing image segmentation quality–concepts, methods and application". In: *Object-based Image Analysis*. Springer, 2008, pp. 769–784.

[80] Y. J. Zhang. "A survey on evaluation methods for image segmentation". In: *Pattern Recognition* 29.8 (1996), pp. 1335–1346. DOI: `10.1016/0031-3203(95)00169-7`.

[81] C. Grimm and D. Pugmire. "Visual interfaces for solids modeling". In: *Proceedings of the 8th Annual ACM Symposium on User interface and Software Technology*. ACM Press, 1995, pp. 51–60.

[82] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000).

[83] E. N. Mortensen and W. A. Barrett. "Intelligent scissors for image composition". In: *Proceedings of the 22nd Annual conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. ACM Press, 1995, pp. 191–198.

[84] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph drawing: Algorithms for the visualization of graphs*. Prentice Hall, 1999.

[85] D. G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision* 60.2 (Nov. 2004), pp. 91–110.

[86] S. Y. Lee, W. Y. Poon, and P. M. Bentler. "A two-stage estimation of structural equation models with continuous and polytomous variables". In: *British Journal of Mathematical and Statistical Psychology* 48 (1995), pp. 339–58.

[87] J. Greulich. *Rekonstruktion von segmentierten Grenzschichten mittels B-Spline Fitting*. Bachelor thesis supervised by Heike Leitte, Institut für Informatik, Ruprecht-Karls-Universität Heidelberg. 2015.

[88] O. Rübel, G. H. Weber, S. V. E. Keränen, C. C. Fowlkes, C. L. L. Hendriks, L. Simirenko, N. Y. Shah, M. B. Eisen, M. D. Biggin, H. Hagen, D. Sudar, J. Malik, D. W. Knowles, and B. Hamann. "PointCloudXplore: Visual analysis of 3d gene expression data using physical views and parallel coordinates". In: *Eurographics Symposium on Visualization Proceedings*. 2006, pp. 203–210. DOI: `10.2312/VisSym/EuroVis06/203-210`.

[89] J. J. Thomas and K. A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Center, 2005.

[90] P. Saraiya, C. North, and K. Duca. "An insight-based methodology for evaluating bioinformatics visualizations". In: *IEEE Transactions on Visualization and Computer Graphics* 11.4 (2005), pp. 443–456. DOI: `10.1109/TVCG.2005.53`.

[91] H. M. Sauro, M. Hucka, A. Finney, C. Wellock, H. Bolouri, J. Doyle, and H. Kitano. “Next generation simulation tools: the systems biology workbench and BioSPICE integration”. In: *OMICS* 7.4 (2003), pp. 355–372. DOI: `10.1089/153623103322637670`.

[92] E. Demir, O. Babur, U. Dogrusoz, A. Gursoy, G. Nisanci, R. Cetin-Atalay, and M. Ozturk. “PATIKA: an integrated visual environment for collaborative construction and analysis of cellular pathways.” In: *Bioinformatics* 18.7 (2002), pp. 996–1003.

[93] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC Press, 1986.

Curriculum Vitae of Julia Portl

Doctoral Studies

May 2011 – Nov 2016	**Doctoral student of Prof. Dr. Heike Leitte at Heidelberg University, Germany and at Technical University Kaiserslautern, Germany**
Thesis	***Visual Analysis of Neuromuscular Junctions***
Sep 2015 – Nov 2016	Researcher in the *Visual Information Analysis* group of Prof. Dr. Heike Leitte at Technical University Kaiserslautern, Germany
Mar 2014 – Feb 2015	Researcher in the *Cryo-Electron Microscopy* group of Prof. Dr. Rasmus Schröder at BioQuant, Heidelberg University, Germany
May 2011 – Aug 2015	Researcher in the *Computer Graphics und Visualization* group of JProf. Dr. Heike Leitte at *IWR (Interdisziplinäres Zentrum für Wissenschaftliches Rechnen – Interdisciplinary Center for Scientific Computing)*, Heidelberg University, Germany
May 2011 – Nov 2016	Scholarship holder and member of the *Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences (HGS MathComp)*, Heidelberg University, Germany

Studies

Nov 2010	Graduation *Erstes Staatsexamen* (Scientific certification as secondary teacher for academic high schools, comparable to a master's degree) at Heidelberg University, Germany
Thesis	*Subdivision Curve and Surface Fitting for Mesh Compression*
2003 – 2010	Studies of Mathematics and Computer Science at Heidelberg University, Germany
Sep 2005 – Dec 2005	Internship as secondary-school teacher at the academic high school Feudenheim-Gymnasium Mannheim, Germany

Kaiserslautern, December 14, 2016 Julia Portl

PUBLICATIONS

UNDER REVIEW (FAST TRACK)

- J. Portl, J. Stegmaier, I. V. Mang, R. R. Schröder, M. Reischl, and H. Leitte. "NeuroMap: Support for the Proofreading and Correction of Surface Segmentations in Image Stacks". In: *Computer Graphics Forum* (2016). Under Review

PUBLISHED PEER-REVIEWED

- J. Stegmaier, N. Peter, J. Portl, I. V. Mang, R. R. Schröder, H. Leitte, R. Mikut, and M. Reischl. "A Framework for Feedback-based Segmentation of 3D-Image Stacks". In: *Current Directions in Biomedical Engineering* 2.1 (Sept. 2016), pp. 437–441. DOI: `10.1515/cdbme-2016-0097`
- J. Portl, M. Reischl, J. Stegmaier, R. R. Schröder, I. V. Mang, and H. Leitte. "Visualization for Error-controlled Surface Reconstruction from Large Electron Microscopy Image Stacks". In: *IEEE Vis: Visualization in Practice*. 2015. URL: `http://julia.portl.space/Papers/portl15a.pdf`
- H. Leitte, J. Portl, I. V. Röder, R. R. Schröder, and I. Wacker. "Towards a Structured Analysis of Quantitative Descriptors from Segmented Biological Image Data". In: *Visualization in Medicine and Life Sciences*. Ed. by L. Linsen, H.-C. Hege, and B. Hamann. The Eurographics Association, 2013. DOI: `10.2312/PE.VMLS.VMLS2013.025-029`

TECHNICAL REPORT

- J. Portl and H. Leitte. *Fold detection in cell membrane segmentations of neuromuscular junctions*. Tech. rep. Revised Version from 2016. 2013. URL: `http://arxiv.org/pdf/1605.09250.pdf`

Colophon

This document was typeset using LaTeX.

The style is based on a template from LaTeXTemplates.com.

The name of the used font is *Palatino*.

Images were created by means of the open-source graphics software
Inkscape and GIMP.

Implementations are in C/C++ using Qt and OpenSceneGraph and in MATLAB.

Algorithms are integrated in the visualization software Scifer.